66 PROGRAMMI PER ZX81 E ZX80 CON NUOVA ROM + HARDWARE

GAETANO MARANO

GRUPPO EDITORIALE JACKSON

66 PROGRAMMI PER ZX81 E ZX80 CON NUOVA ROM + HARDWARE

di

Gaetano Marano

Gruppo
Editoriale
Jackson
Via Rosellini, 12
20124 Milano

SOMMARIO

PREFAZIONE

Questo libro contiene 66 programmi di vario genere, scritti per girare sul computer Sinclair ZX81, computer che, per le sue eccezionali qualità unite ad un modestissimo prezzo, ha già conquistato centinaia di migliaia di persone in tutto il mondo riuscendo a superare già, secondo gli ultimi dati forniti dalla Sinclair, in numero di pezzi, le vendite di qualsiasi altro modello di computer, singolarmente preso, costruito sino ad oggi.
LO ZX81 viene utilizzato anche troppo per realizzare dei giochi. Ho pensato, perciò, di offrire in questo libro una maggiore varietà di programmi cercando di sfruttare nella gran parte di essi soprattutto le capacità grafiche dello ZX81 e, comunque, senza dimenticare del tutto i giochi che sono, naturalmente, presenti nella parte finale di questo libro con 9 programmi.
Oltre alle normali possibilità dello ZX81, il lettore troverà alcuni circuiti molto semplici e poco costosi che consentono di aggiungere nuove possibilità allo ZX81. Oltre ad un utilissimo avvisatore acustico per la tastiera (realizzabile con meno di 3.000 lire) e ad un alimentatore con accumulatore (per conservare i dati della memoria e usare il computer anche durante le interruzioni della corrente di rete), tali circuiti sono costituiti soprattutto da un circuito di INTERFACCIA che costa in componenti meno di 5000 lire e che permette di collegare lo ZX81 a qualsiasi dispositivo esterno e da un circuito MUSICALE realizzabile con circa 10000 lire e che fornisce tramite semplici istruzioni in BASIC, fino a 50 note su 4 ottave.
Di questi due ultimi circuiti, le cui applicazioni sono infinite, non vengono forniti solo gli schemi e le informazioni per costruirli, ma anche numerosi programmi pratici.
In particolare la scheda musicale è utilizzata per vari effetti sonori, per realizzare un organo a 4 ottave che usa i tasti dello ZX81 e per dare il sonoro a vari programmi tra cui naturalmente alcuni dei giochi che diventano così probabilmente gli unici giochi con effetti sonori disponibili a tutt'oggi per i computers Sinclair.
Un altra novità di questo libro è l'uso in alcuni programmi (come per esempio l'organo elettronico) di speciali tastiere disegnate da sovrapporre alla tastiera sensitiva degli ZX per assegnare ai tasti alcune funzioni particolari.
Come indicato dal titolo, anche chi ha il vecchio ZX80 con la nuova ROM, può utilizzare molti dei programmi di questo libro compresi i programmi che usano i circuiti di interfaccia e musicale, circuiti che naturalmente possono essere collegati anche allo ZX80.

Gaetano Marano

NOTE IMPORTANTI SUI PROGRAMMI E SULLA RAM UTILIZZATA

Tutti i programmi possono girare sullo ZX81 ed una gran parte anche sullo ZX80 con la nuova ROM.
A fianco del titolo di ciascun programma viene indicato se il programma stesso funziona in FAST e/o SLOW. Se c'è scritto FAST il programma può girare su ZX80 e su ZX81, naturalmente in modo FAST. Se c'è scritto SLOW il programma può girare solo sullo ZX81. Se, infine, c'è scritto FAST o SLOW il programma può girare sia su ZX81 che su ZX80, però nel caso dello ZX80 si hanno alcune limitazioni, come per esempio in un disegno, l'impossibilità di vederlo formarsi, in sostanza, quindi, si ha la normale elaborazione ma non l'animazione grafica.
Per quanto riguarda il fabbisogno di memoria, solo per alcuni programmi, come per esempio gli effetti sonori, è sufficiente il singolo KBYTE e, comunque, nessuno richiede più di 4 KBYTES, per cui possono andare bene sia gli ZX espansi con la cartuccia RAM da 16K che con la cartuccia RAM da 3K. Non solo, ma in alcuni

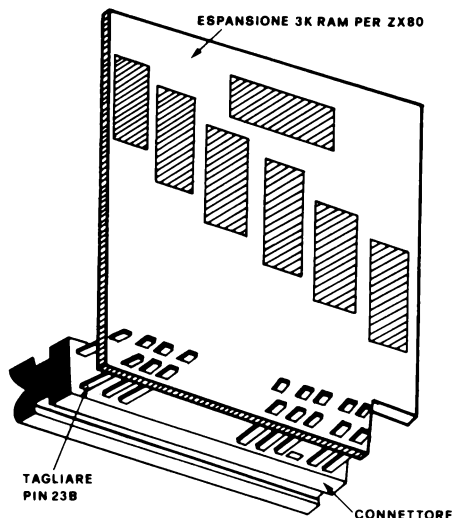



Figura 1

programmi che hanno delle routines in linguaggio macchina, le routines stesse sono state inserite entro i primi 4K di RAM in modo che tali programmi possano essere usati, senza modifiche, sia da chi dispone di RAM da 4K che da chi dispone di RAM da 16K.
Infine, se avete solo lo ZX81, ma non l'espansione RAM da 16K e volete usare la vecchia espansione da 3K, questo è possibile ma occorre effettuare sulla cartuccia da 3K una piccola modifica interrompendo e piegando in modo che non faccia contatto, il collegamento tra il connettore ed il circuito stampato, corrispondente a Pin 23B del connettore stesso, come visibile in fig. 1.

AVVISATORE ACUSTICO PER TASTIERA ED ALIMENTATORE TAMPONE PER ZX80 E ZX81

Avvisatore Acustico

L'avvisatore acustico per la tastiera sensitiva degli ZX80/81 produce un breve suono ogni volta che viene premuto uno dei tasti.
L'utilità è evidente poichè questo semplice circuito (che può essere realizzato con 3-4.000 lire) permette di inserire programmi anche molto lunghi in un tempo molto più breve e senza dovere guardare il video ogni volta che si preme un tasto per verificare l'avvenuto inserimento della lettera, del simbolo o della Keyword richiesta.
Il circuito è in fig. 2 ed utilizza come semiconduttori due integrati ed un transistor.

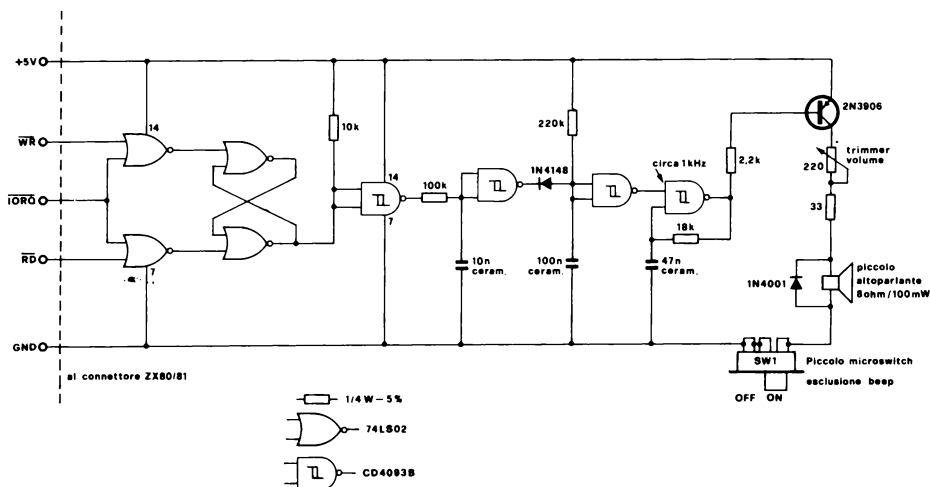



Figura 2

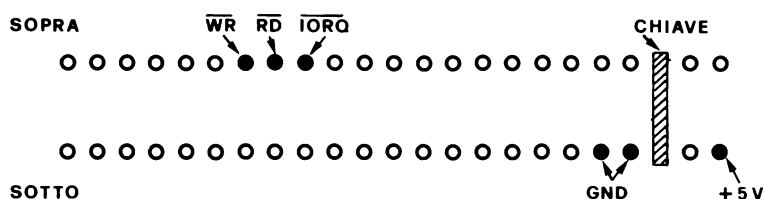
Tutti i componenti possono essere montati su un pezzettino di circuito stampato forato facendo, naturalmente, attenzione a collegare esattamente i vari componenti, tra questi soprattutto i due diodi che occorre verificare che siano collegati come dallo schema, prima di dare corrente.
Il circuito dell'avvisatore acustico va collegato tramite 5 fili (come visibile sempre in fig. 2) o all'interno del computer o, preferibilmente, al connettore dell'espansione di memoria tramite 5 spezzoni di filo sottile ad un capo di diverso colore, come per esempio il filo per WIRE-WRAP.
In fig. 3 è visibile il connettore della RAM, per ZX80/81, visto da dietro e con indicati i 5 Pin ai quali vanno saldati i 5 fili dell'avvisatore acustico (+5V, GND, $\overline{WR}$, $\overline{RD}$, $\overline{IORQ}$).

Connettore RAM 3K/16K per ZX80/81 visto da dietro

Figura 3

Nel collegare i fili al connettore fare molta ATTENZIONE a saldare i fili stessi ai Pin indicati e non ad altri e, soprattutto, controllare di non avere cortocircuitato durante la saldatura qualcuno dei Pin adiacenti, SPECIALMENTE i Pin che portano i +5V e i +9V.
TENETE PRESENTE CHE UN'ERRATA COSTRUZIONE DEL CIRCUITO O UNA ERRATA SALDATURA DEI FILI AL CONNETTORE PUO' CAUSARE DANNI AL COMPUTER.
Per cui se vi occupate di programmazione ma non di circuiti integrati, è preferibile che facciate realizzare o controllare il tutto da un tecnico che abbia già esperienza di montaggi e saldature.
In fig. 4 vi sono le configurazioni degli integrati e del transistore usati.
Una cosa molto importante da dire prima di terminare questa parte è che l'avvisatore acustico genera un suono alla pressione dei tasti dello ZX81 solo quando lo ZX81 funziona nel modo FAST; questa non è una limitazione perchè, per inserire velocemente dei programmi sullo ZX81 si deve, comunque operare in FAST, poichè come avete avuto modo di verificare l'inserimento di nuove linee con il computer in SLOW è già di per sè molto lento.

L'avvisatore acustico dispone anche di un piccolo interruttore che serve per disinserire il suono quando non è necessario, come per esempio quando è in funzione anche la scheda musicale che verrà esposta più avanti.

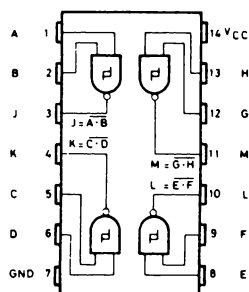
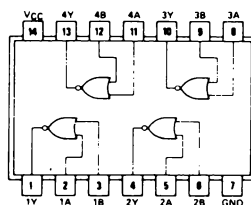
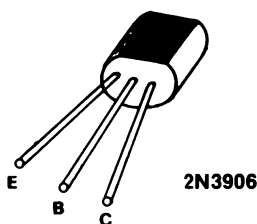



Figura 4

Alimentatore tampone

Questo alimentatore oltre alla funzione secondaria di fornire una tensione maggiormente stabilizzata, ha lo scopo principale di preservare i dati scritti nella RAM durante le interruzioni della tensione di rete e questo per un tempo più che sufficiente a conservare il programma (che a volte può avere richiesto molto tempo per esere scritto) su cassetta e, nella maggior parte dei casi, anche più che sufficiente ad attendere il ritorno della luce.

LM317T
Vin
Vout
ADJ
IC1
Io
Ra
$I_o = \frac{1,2}{R_a}$
LM317T
Vi
Vo
IC2
Ra
Rb
$V_o = 1,25 \left(1 + \frac{R_b}{R_a}\right)$
Interruttore 2vie – 2posizioni
220V ~
12V
Trasformatore
220V/12V/2A
Ponte 100V/2 – 3A
Diodo 100V/3A
off
on
IC1 LM317T
4
1W
Diodo
100V/3A
680
2200µ
35V
LED rosso
Io = 300mA
Accumulatore
ermetico al piombo
12V/3Ah
GND
1/4W – 5%
220µ
35V
IC2 LM317T
120
33µ
16V
1,5k
1,5k
330
LED
giallo
Spina jack
Ø 3,5mm
+9V circa
GND

Figura 5

L'utilizzo ideale dell'alimentatore tampone è insieme ad un piccolo televisore anch'esso fornito di alimentazione autonoma a batteria.
Il circuito dell'alimentatore tampone è in fig. 5. La tensione necessaria durante i periodi di blackout è fornita da un accumulatore ermetico al piombo da 12V 3Ah (B_1).

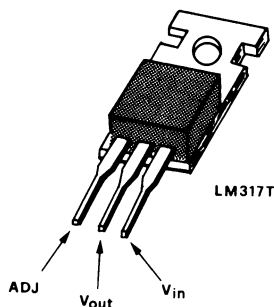



Figura 6

La configurazione dei due LM317T, utilizzati nel circuito, è in fig. 6; di questi due integrati uno (IC_1) viene utilizzato per ricaricare l'accumulatore e l'altro (IC_2) per stabilizzare la tensione ai circa 9V necessari per il computer.
La fig. 7 mostra una possibile realizzazione pratica dell'alimentatore. Anche per questo circuito vale quanto detto per l'avvisatore acustico circa la necessità di fare attenzione ed effettuare tutti i collegamenti nel modo giusto.

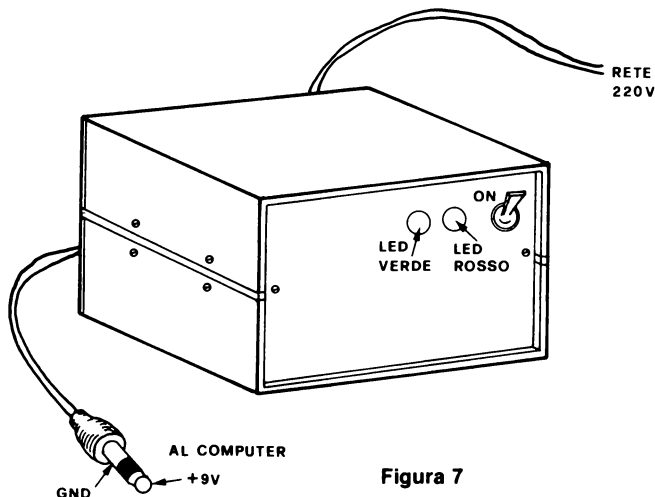



Figura 7

I due LM317T vanno montati su due piccoli dissipatori oppure sul retro del mobile metallico, in questo caso pero' isolandoli tra loro con delle miche e verificando tale isolamento con l'ohmmetro PRIMA di dare corrente all'apparecchio.
In particolare è MOLTO IMPORTANTE CONTROLLARE, PRIMA DI COL-

LEGARE L'ALIMENTATORE AL COMPUTER, che la polarità del JACK di uscita sia esattamente uguale a quella dell'alimentatore Sinclair con, quindi, il positivo (+9V) sulla punta, ed inoltre controllare che la tensione di uscita non sia superiore a 11-12V.

GRAFICA E DISEGNI

Lavagna grafica — SLOW

Questo primo programma (fig. 8) serve per realizzare e modificare dei disegni sul video utilizzando le istruzioni PLOT-UNPLOT.
Il disegno viene realizzato premendo alcuni dei tasti del computer. Per evitare confusione nel premere i tasti viene utilizzata in questo programma la tastiera N. 1 inserita alla fine del libro. Tale tastiera va ritagliata come indicato e sovrapposta alla tastiera dello ZX81.

Figura 8

```
  10 REM  LAVAGNA GRAFICA
 100 SLOW
 110 LET X=0
 120 LET Y=0
 130 LET T=0
 140 IF X<63 THEN LET X=X+(INKEY
$="7")+(INKEY$="U")+(INKEY$="H")
 150 IF X>0 THEN LET X=X-(INKEY$
="5")-(INKEY$="R")-(INKEY$="F")
 160 IF Y<43 THEN LET Y=Y+(INKEY
$="5")+(INKEY$="6")+(INKEY$="7")
 170 IF Y>0 THEN LET Y=Y-(INKEY$
="F")-(INKEY$="G")-(INKEY$="H")
 180 IF INKEY$="P" THEN GOTO 300
 190 IF INKEY$="O" THEN STOP
 200 IF INKEY$="9" THEN LET T=0
 210 IF INKEY$="0" THEN LET T=1
 220 PLOT X,Y
 230 LET A=X
 240 LET B=Y
 250 IF T=1 THEN UNPLOT A,B
 260 GOTO 140
 300 SCROLL
 310 IF NOT INKEY$="P" THEN GOTO
140
 320 GOTO 300
```

Figura 9

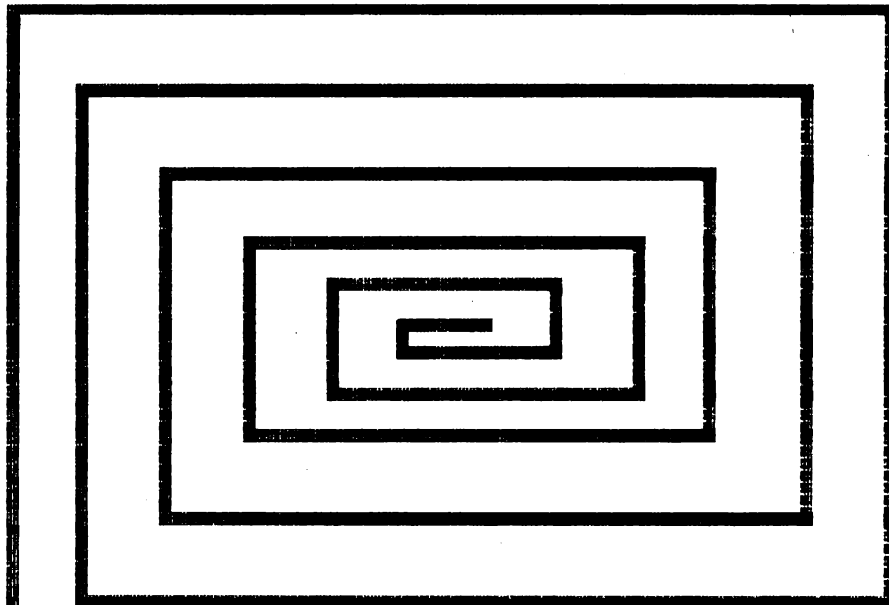



Figura 10

All'inizio, subito dopo RUN e NEWLINE appare un piccolo quadratino nero in basso a sinistra dello schermo ed il programma è predisposto per disegnare. Premendo uno degli 8 tasti rossi il quadratino si sposta nella direzione indicata sul tasto premuto lasciando dove passa una traccia.
Se si vuole spostare il quadratino per andare a disegnare in un altra parte dello schermo o se si vuole effettuare una modifica al disegno già fatto, si può premere il tasto verde "MODIFICHE E SPOSTAMENTI" e quindi i tasti rossi necessari per spostarsi.
Una volta effettuata la modifica o lo spostamento si può ritornare a disegnare premendo il tasto verde "DISEGNO" e quindi i tasti rossi.
Il tasto verde "SCROLL" permette di spostare tutto il disegno fatto fino a quel

```
  10 REM  LAVAGNA
  20 REM  ALFANUMERICA
  30 REM  E GRAFICA
 100 SLOW
 110 LET L=0
 120 LET C=0
 130 LET T=0
 140 LET X=0
 145 LET J=0
 150 LET K=CODE INKEY$
 155 IF X<64 AND K=116 THEN LET
J=1
 156 IF K=222 THEN LET J=0
 157 IF K=218 THEN LET X=128
 158 IF K=218 THEN LET J=0
 160 IF K>0 AND K<64 THEN LET X=
K
 170 IF C<31 AND K=115 THEN LET
C=C+1
 180 IF C>0 AND K=114 THEN LET C
=C-1
 190 IF L<21 AND K=113 THEN LET
L=L+1
 200 IF L>0 AND K=112 THEN LET L
=L-1
 210 IF K=117 THEN SCROLL
 220 IF K=223 THEN LET T=0
 230 IF K=119 THEN LET T=1
 240 IF J=0 THEN PRINT AT L,C;CH
R$ X
 250 IF J=1 THEN PRINT AT L,C;CH
R$ (X+128)
 260 IF T=1 THEN PRINT AT L,C;"
"
 270 GOTO 150
```

Figura 11

momento verso l'alto. Infine il tasto verde "STOP" blocca l'esecuzione del programma.
Le figg. 9 e 10 mostrano due dei molti tipi di disegni realizzabili con il programma.

Lavagna alfanumerica e grafica SLOW

Questo programma (fig. 11), a differenza di quello precedente, usa per disegnare l'istruzione PRINT AT e ciò permette di inserire nel disegno tutti i simboli grafici, le lettere, i numeri, ecc. anche in campo inverso.
Per i vari comandi viene utilizzata la tastiera N. 2 che, a differenza della tastiera utilizzata nel programma precedente, non viene sovrapposta alla tastiera dello ZX81 ma va sistemata al di sopra della prima fila di tasti (1....Ø); questo perchè i tasti vanno lasciati liberi per inserire i caratteri con cui realizzare i vari disegni.
Con il programma in funzione i vari comandi della tastiera speciale vanno azionati premendo contemporaneamente il tasto SHIFT ed uno dei tasti della prima fila.
Premendo uno qualsiasi dei tasti (senza SHIFT) appare sul video il carattere o numero corrispondente, se si vuole lo stesso carattere ma inverso occorre premere SHIFT e il tasto 9 escluso lo spazio inverso che si ottiene premendo SHIFT e il tasto 2; per ritornare ai caratteri normali si preme SHIFT ed il tasto 3.
Una volta scelto il carattere si può iniziare a disegnare premendo SHIFT e i tasti 5, 6, 7, 8, oppure lo si può spostare usando gli stessi tasti ma premendo prima SHIFT

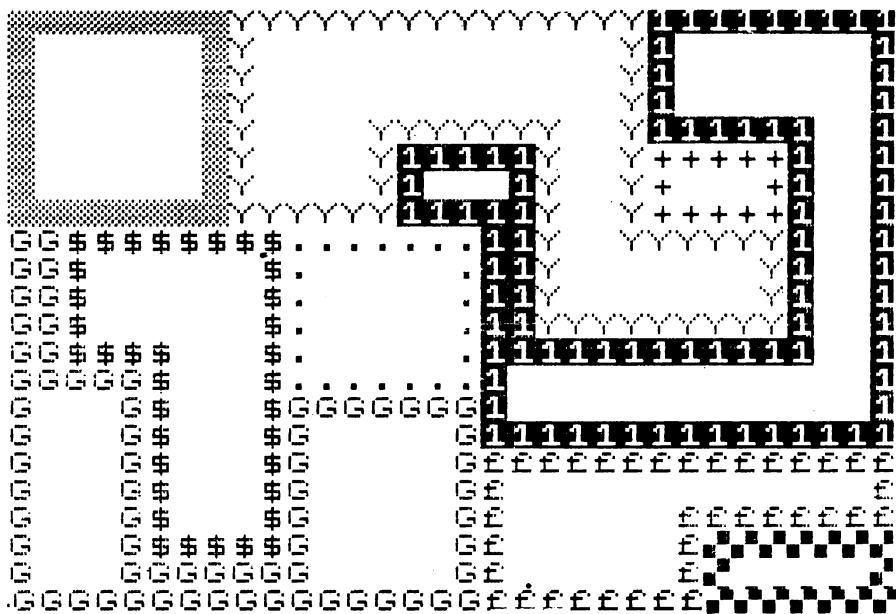



Figura 12

```
  10 REM  RODITORE
 100 SLOW
 110 FOR A=1 TO 88
 120 PRINT "████████";
 130 NEXT A
 140 PRINT
 150 LET L=21
 160 LET C=0
 170 PRINT AT L,C;"."
 180 FOR T=1 TO 3
 190 NEXT T
 200 PRINT AT L,C;" "
 210 IF C>0 AND INKEY$="5" THEN
LET C=C-1
 220 IF L<21 AND INKEY$="6" THEN
 LET L=L+1
 230 IF L>0 AND INKEY$="7" THEN
LET L=L-1
 240 IF C<31 AND INKEY$="8" THEN
 LET C=C+1
 250 GOTO 170
```

Figura 13

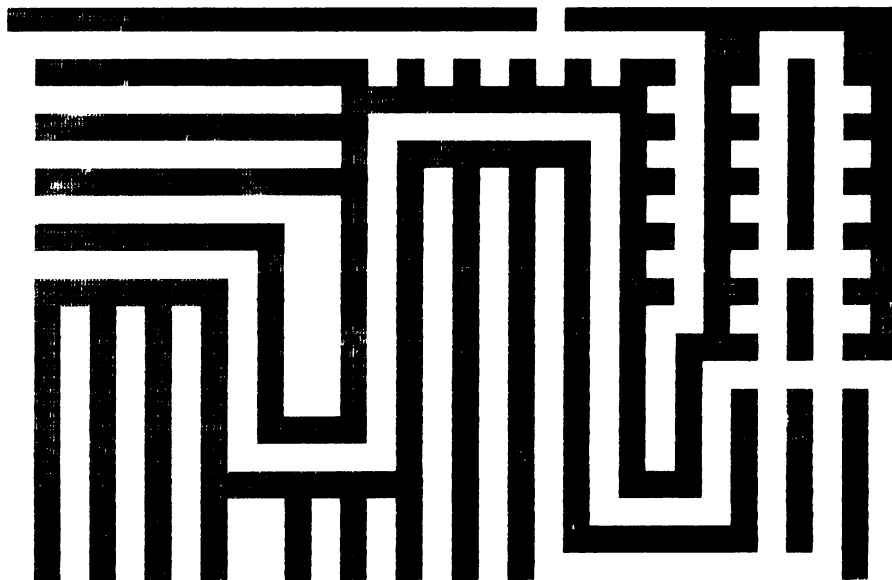



Figura 14

ed il tasto 0, tasto che si deve premere anche per effettuare delle modifiche al disegno.
Effettuata la modifica o lo spostamento, per ritornare alla scrittura si deve premere SHIFT ed il tasto 4.
La fig. 12 mostra un esempio di disegno realizzato con il programma appena esposto.

Roditore SLOW

Questo programma grafico (fig. 13) inizia riempendo lo schermo di spazi inversi o di qualsiasi altro carattere, numero o simbolo grafico.
Dopo di ciò appare nell'angolo in basso a sinistra un punto lampeggiante. Tale punto può essere spostato nelle quattro direzioni tramite i tasti 5, 6, 7, 8.
Dove passa il punto "mangia" una parte del rettangolo di caratteri o spazi inversi formatosi all'inizio del programma.

```
120 PRINT "▒▒▒▒▒▒▒▒";
```

Figura 15

Un esempio di disegno realizzabile con il programma "Roditore" è una specie di labirinto visibile in fig. 14. La linea 120 contiene tra gli apici 8 spazi inversi che con le linee 110 e 130 producono sullo schermo un rettangolo nero.

```
120 PRINT "$$$$$$$$";
```

Figura 16

Per avere un diverso fondo la linea 120 può essere modificata come visibile nelle figure 15 - 16 - 17. In queste nuove linee i caratteri tra gli apici sono sempre 8 però non si tratta più di spazi inversi ma del carattere grafico del tasto A (fig. 15) o del

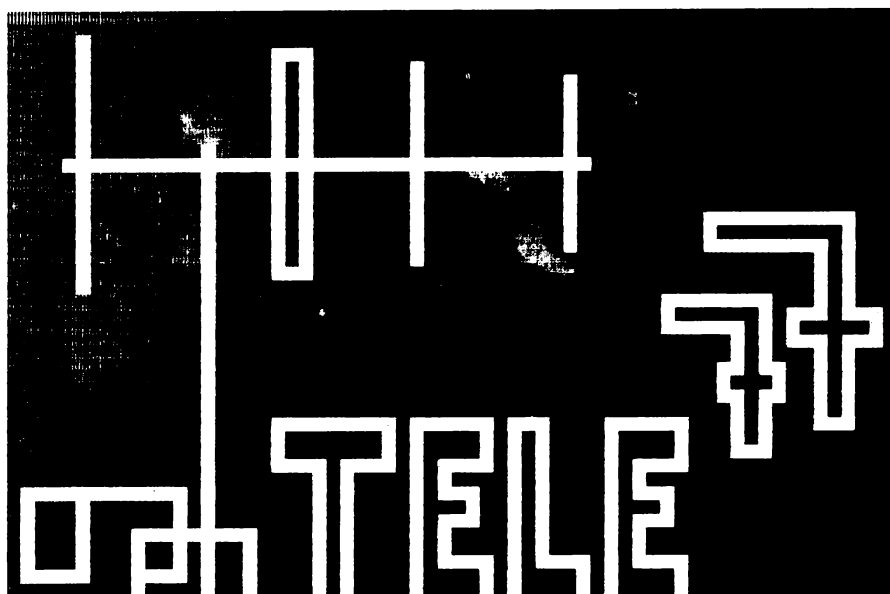
Figura 17

simbolo inverso del $ (fig. 16) o ancora di una serie di punti inversi (fig. 17).
Nel listato di fig. 13 si possono eliminare, eventualmente, le istruzioni 180 e 190 per rendere più veloce il movimento del punto.
Il programma può essere fermato con il tasto BREAK.

```
  10 REM  RODITORE  -2-
  20 REM  CON PLOT-UNPLOT
 100 SLOW
 110 FOR A=1 TO 88
 120 PRINT "████████";
 130 NEXT A
 140 PRINT
 150 LET X=0
 160 LET Y=0
 165 LET K=0
 170 IF K=0 THEN PLOT X,Y
 180 UNPLOT X,Y
 190 IF K=1 THEN PLOT X,Y
 200 IF INKEY$="S" THEN LET K=1
 205 IF INKEY$="M" THEN LET K=0
 210 IF X>0 AND INKEY$="5" THEN
LET X=X-1
 220 IF Y<43 AND INKEY$="7" THEN
 LET Y=Y+1
 230 IF Y>0 AND INKEY$="6" THEN
LET Y=Y-1
 240 IF X<63 AND INKEY$="8" THEN
 LET X=X+1
 250 GOTO 170
```

Figura 18

Figura 19

Roditore -2- con PLOT-UNPLOT — SLOW

Questo secondo programma di Roditore grafico (fig. 18) è simile al precedente con la differenza che usa le istruzioni PLOT e UNPLOT, il che permette di realizzare dei disegni con lo stesso sistema ma con una definizione quattro volte maggiore, inoltre è anche possibile modificare un disegno già fatto o correggere eventuàli errori.
L'istruzione 120 contiene tra gli apici 8 spazi inversi, tale istruzione assieme alla 110 e 130 disegnano un grosso rettangolo nero sullo schermo, dopodichè il quadratino in basso a sinistra di tale rettangolo lampeggia.
Tramite i tasti 5, 6, 7, 8 si può disegnare spostando il quadratino lampeggiante nelle quattro direzioni.
Se si vuole modificare un disegno o si vuole spostare il quadratino lampeggiante in una zona del rettangolo scuro, si può premere prima il tasto "S" (Scrive) premendo poi il tasto "M" (Mangia) per tornare a disegnare.
La fig. 19 mostra un esempio di disegno realizzato con questo programma.

Programma grafico — FAST O SLOW

Il programma di fig. 20 può essere utilizzato per tracciare il grafico di una qualsiasi funzione matematica.

```
  10 REM  PROGRAMMA GRAFICO
  20 REM ---------------------
  77 PRINT TAB 4;"ATTENDERE 2 MI
NUTI"
  88 PAUSE 200
  99 CLS
 100 FAST
 110 DIM A(64)
 120 FOR K=1 TO 64
 130 LET A(K)=K*K/110
 140 NEXT K
 150 SLOW
 160 FOR B=0 TO 63
 170 GOSUB 210
 180 NEXT B
 190 FAST
 200 STOP
 210 FOR E=B TO 63
 220 PLOT E,A(B+1)
 230 NEXT E
 240 RETURN
```

Figura 20

La fig. 21, per esempio, mostra il grafico relativo alla funzione matematica presente nella linea 130 del programma di fig. 20 (K*K/11Ø). Occorre tenere presente che per disegnare tale grafico sullo schermo il computer ha bisogno di circa 2 minuti di tempo.

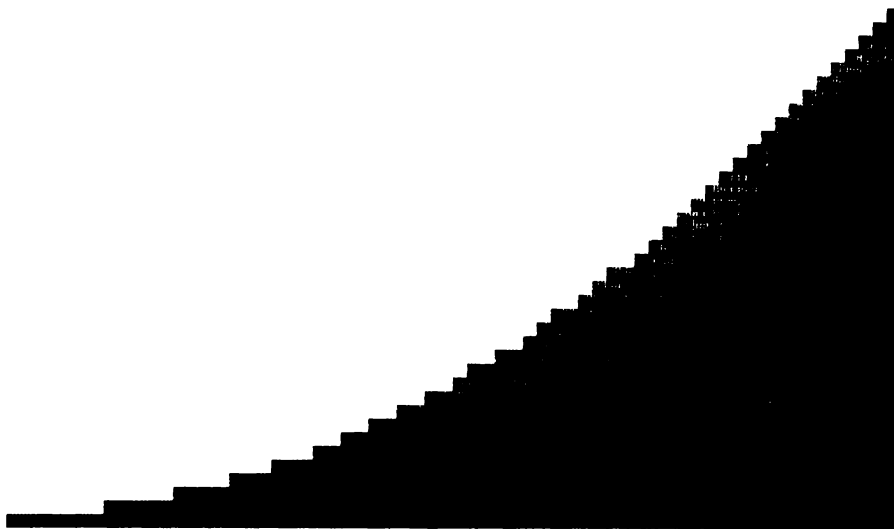



Figura 21

Se si vuole disegnare solo la curva della funzione matematica senza la parte scura di riempimento ed anche con una maggiore velocità di esecuzione, si può modificare il programma di fig. 20 sostituendo la linea 17Ø GOSUB 21Ø con la linea 170 indicata in fig. 22.

```
170 PLOT B,A(B+1)
```

Figura 22

```
130 LET A(K)=RND*(K/1.5)
```

Figura 23

```
130 LET A(K)=TAN K*5
```

Figura 24

```
130 LET A(K)=LN K*10
```

Figura 25

Nelle figure 23, 24, 25 vengono date altre tre versioni della linea 130 (da inserire nel programma di fig. 20) con tre diverse funzioni matematiche per ottenere altrettante diverse curve sullo schermo.

Programma universale per PRINT AT

FAST O SLOW

Molto spesso all'inizio di un programma è necessario per prima cosa inserire in varie parti dello schermo tutta una serie di lettere, numeri e simboli, grafici e non, per formare delle frasi e/o dei disegni.
Questa operazione può richiedere anche molte decine di linee di programma contenenti l'istruzione PRINT AT.

```
  10 REM  PROGRAMMA UNIVERSALE
  20 REM  PER PRINT AT
  30 REM  -IN FAST O SLOW-
 100 SLOW
 110 REM  CODICI LINEE IN L$
 120 LET L$="00123333456663444444
55555777888999AAAAAAABBBCCCDDD"
 130 REM  CODICI COLONNE IN C$
 140 LET C$="010KJKLMKKKLMTPQRST
PQRST567468369345678936946 8567"
 150 REM CARATTERI IN X$
 160 LET X$="▘CLTTESTST     APRINT
▒▒▒▒▒▒▒▒▒ ........................"
 170 IF LEN L$<>LEN C$ OR LEN C$
<>LEN X$ THEN GOTO 220
 180 FOR P=1 TO LEN X$
 190 PRINT AT CODE L$(P TO )-28,
CODE C$(P TO )-28;X$(P TO P)
 200 NEXT P
 210 STOP
 220 PRINT AT 0,10;" ERRORE "
 230 PAUSE 100
 240 FAST
 250 LIST
```

Figura 26

Se però si stabilisce prima con esattezza quali caratteri vanno inseriti e per ciascuno in quale linea e colonna, tutta questa fase può essere svolta (in FAST o SLOW) dal programma di fig. 26.
I caratteri da inserire sono nella variabile X$ mentre le linee e le colonne in cui inserire tali caratteri sono contenuti in forma codificata nelle variabili L$ (linea) e C$ (colonna).

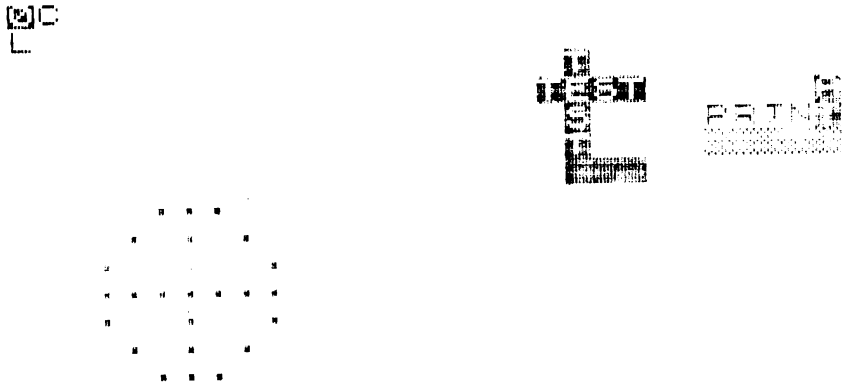
Tabella 1

CODICI LINEE E COLONNE

L/C		COD.	L/C		COD.	C		COD.
0	>	0	11	>	B	22	>	M
1	>	1	12	>	C	23	>	N
2	>	2	13	>	D	24	>	O
3	>	3	14	>	E	25	>	P
4	>	4	15	>	F	26	>	Q
5	>	5	16	>	G	27	>	R
6	>	6	17	>	H	28	>	S
7	>	7	18	>	I	29	>	T
8	>	8	19	>	J	30	>	U
9	>	9	20	>	K	31	>	V
10	>	A	21	>	L			

I codici delle linee e delle colonne sono indicate nella tabella 1. Se per esempio il primo carattere di L$ è "G", il primo carattere di C$ è "T" e il primo carattere di X$ è "+", il programma piazzerà il simbolo+ alla linea 16 colonna 29.
Nel programma di fig. 26 è inserito in L$, C$ e X$ un esempio che produce il disegno di fig. 27.
Naturalmente L$, C$ e X$ devono avere la stessa lunghezza.
Nel programma sono state, perciò, inserite alcune istruzioni (170, 220, 230, 240, 250) che segnalano con la scritta "ERRORE" eventuali differenze di lunghezza.

Figura 27

Programma universale per PLOTTAGGIO — FAST O SLOW

Come nel caso precedente per PRINT AT, anche una lunga serie di linee di programma con PLOT può essere condensata in un programma più breve come è visibile in fig. 28.
In tale programma le coordinate X e Y di ciascuno punto da plottare sono inserite

```
  10 REM  PROGRAMMA UNIVERSALE
  20 REM  PER PLOTTAGGIO
  30 REM  - IN FAST O SLOW -
 100 SLOW
 110 REM COORDINATE ORIZZ. IN X$
 115 REM ------------------------
 120 LET X$="13 14 15 16 17 18 1
9 20 21 22 23 23 23 24 24 24 24
23 23 23 22 21 20 19 18 17 16 15
 14 13 12 11 10 09 08 07 06 05 0
6 05 05 05 05 06 06 06 07 08 09
10 11 12 20 19 18 17 16 15"
 125 REM ------------------------
 130 REM COORDINATE VERT.IN Y$
 135 REM ------------------------
 140 LET Y$="05 05 05 05 06 06 0
6 07 08 09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 23 23 24 24
 24 24 23 23 23 22 21 20 19 18 1
7 16 15 14 13 12 11 10 09 08 07
06 06 06 09 10 11 12 13 14 "
 150 REM ------------------------
 160 FOR P=1 TO LEN X$ STEP 3
 170 PLOT VAL X$(P TO P+1),VAL Y
$(P TO P+1)
 180 NEXT P
 300 PRINT AT 16,8;"A"
```

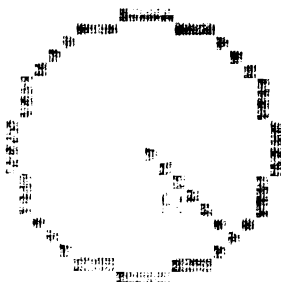



Figura 28

Figura 29

rispettivamente nelle variabili X$ e Y$. Come è visibile tra un numero e l'altro va inserito uno spazio.
Naturalmente le coordinate vanno comprese tra 00 e 63 in X$ e tra 00 e 43 in Y$; i numeri inferiori a 10 vanno inseriti con uno 0 davanti (per es. il numero 7 va inserito come 07).
Le coordinate inserite come esempio nel programma di fig. 28 danno il disegno di fig. 29; la lettera R viene aggiunta a disegno ultimato dall'istruzione contenuta nella linea 300. Realizzare lo stesso disegno senza questo programma avrebbe richiesto 58 linee di programma con PLOT.

Plottaggio QUADRATI

FAST O SLOW

Quando occorre disegnare varie figure quadrate si può utilizzare il programma di fig. 30 che disegna quadrati di qualsiasi dimensione inserendo le coordinate X e Y

```
  10 PRINT "INSERIRE X"
  20 INPUT X
  30 CLS
  40 PRINT "INSERIRE Y"
  50 INPUT Y
  60 CLS
  70 PRINT "INSERIRE L"
  80 INPUT L
  90 CLS
 100 GOSUB 4000
 200 STOP
 999 REM ----------------------

4000 REM PLOTTAGGIO QUADRATI
4010 SLOW
4020 FOR A=X TO X+L
4030 IF A<0 OR A>63 THEN GOTO 40
60
4040 IF Y>=0 AND Y<=43 THEN PLOT
 A,Y
4050 IF Y+L>=0 AND Y+L<=43 THEN
PLOT A,Y+L
4060 NEXT A
4070 FOR A=Y TO Y+L
4080 IF A<0 OR A>43 THEN GOTO 41
10
4090 IF X>=0 AND X<=63 THEN PLOT
 X,A
4100 IF X+L>=0 AND X+L<=63 THEN
PLOT X+L,A
4110 NEXT A
4120 RETURN
```

Figura 30

(anche negative) dell'angolo in basso a sinistra del quadrato stesso e naturalmente il lato "L" (vedere fig. 31).

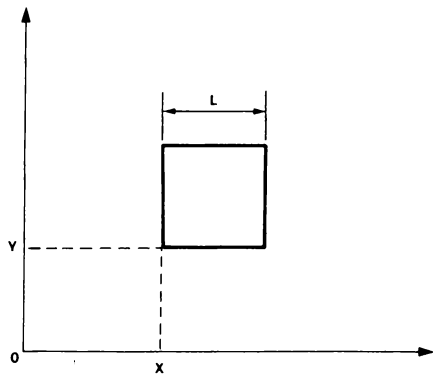



Figura 31

```
 10 REM PLOTT. CASUALE QUADRATI
 20 REM .........................
 30 RAND
 40 FOR Q=1 TO 7
 50 LET X=INT (50*RND)
 60 LET L=5+INT (25*RND)
 70 LET Y=INT (30*RND)
 80 GOSUB 4000
 90 NEXT Q
100 STOP
999 REM -------------------------
```

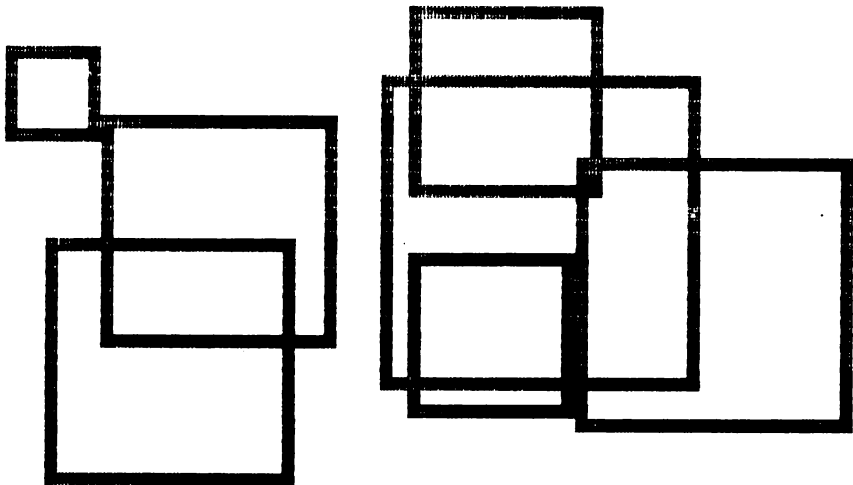



Figura 32

Figura 33

Come esempio si possono sostituire le istruzioni da 10 a 999 in fig. 30 con le nuove istruzioni esposte in fig. 32 per produrre sullo schermo il disegno di 7 quadrati con posizione e grandezza casuali (fig. 33).

Plottaggio CERCHI

FAST O SLOW

È possibile disegnare anche dei cerchi usando il programma di fig. 34. In tale programma occorre inserire all'inizio le coordinate X e Y (anche negative) del

```
  10 PRINT "INSERIRE X"
  20 INPUT X
  30 CLS
  40 PRINT "INSERIRE Y"
  50 INPUT Y
  60 CLS
  70 PRINT "INSERIRE R"
  80 INPUT R
  90 CLS
 100 GOSUB 5000
 200 STOP
 999 REM -----------------------

5000 REM  PLOTTAGGIO CERCHI
5010 SLOW
5020 FOR C=0 TO 360 STEP 1+INT
30/R)
5030 LET K=C*PI/180
5040 LET A=R*COS K+X
5050 LET B=R*SIN K+Y
5060 IF A>=0 AND A<=63 AND B>=0
AND B<=43 THEN PLOT A,B
5070 NEXT C
5080 RETURN
```

Figura 34

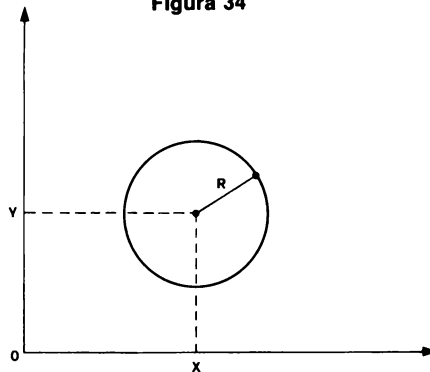



Figura 35

```
 10 REM  CERCHI CONCENTRICI
 20 REM ......................
 30 LET X=32
 40 LET Y=22
 50 FOR R=5 TO 30 STEP 5
 60 GOSUB 5000
 70 NEXT R
100 STOP
999 REM ----------------------
```

Figura 36

centro ed il raggio "R" del cerchio da disegnare (vedere fig. 35). Anche in questo caso come esempio vengono date in fig. 36 alcune nuove istruzioni da inserire nel programma di fig. 34 in sostituzione delle prime 12 istruzioni (da 10 a 999) per ottenere una serie di cerchi concentrici (fig. 37).

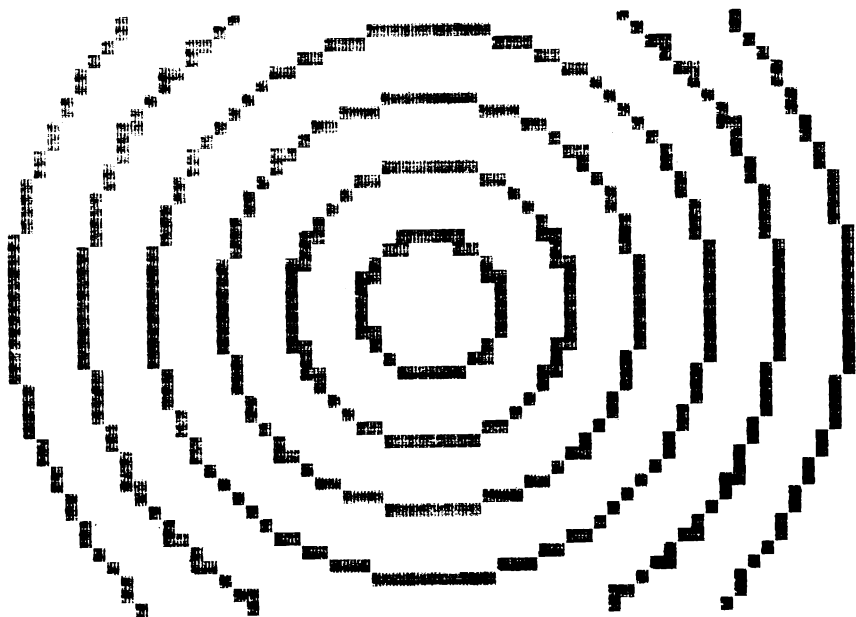



Figura 37

Plottaggio ELLISSI

FAST O SLOW

Per terminare questa breve serie di figure geometriche da disegnare con il computer viene dato anche un programma per disegnare le ellissi (fig. 38), a partire dalle

```
  10 PRINT "INSERIRE X"
  20 INPUT X
  30 CLS
  40 PRINT "INSERIRE Y"
  50 INPUT Y
  60 CLS
  70 PRINT "INSERIRE RMA"
  80 INPUT RMA
  90 CLS
 100 PRINT "INSERIRE RMI"
 110 INPUT RMI
 120 CLS
 150 GOSUB 6000
 200 STOP
 999 REM -----------------------
6000 REM  PLOTTAGGIO ELLISSI
6010 SLOW
6020 IF RMA>=RMI THEN LET S=RMA
6030 IF RMI>=RMA THEN LET S=RMI
6040 FOR C=0 TO 360 STEP 1+INT (
30/S)
6050 LET K=C*PI/180
6060 LET A=(RMA-0.5)*(1+COS K)-R
MA+X
6070 LET B=(RMI-0.5)*(1+SIN K)-R
MI+Y
6080 IF A>=0 AND A<=63 AND B>=0
AND B<=43 THEN PLOT A,B
6090 NEXT C
6100 RETURN
```

Figura 38

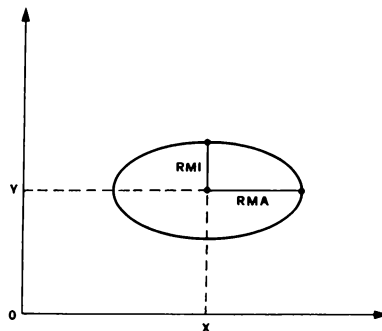



Figura 39

coordinate X e Y (anche negative) del centro e dei valori del raggio maggiore (RMA) e dal raggio minore (RMI) come visibile in fig. 39.
In figura 40 e figura 41 sono mostrati due tipi diversi di ellissi con indicati in alto i valori inseriti per ottenerle.

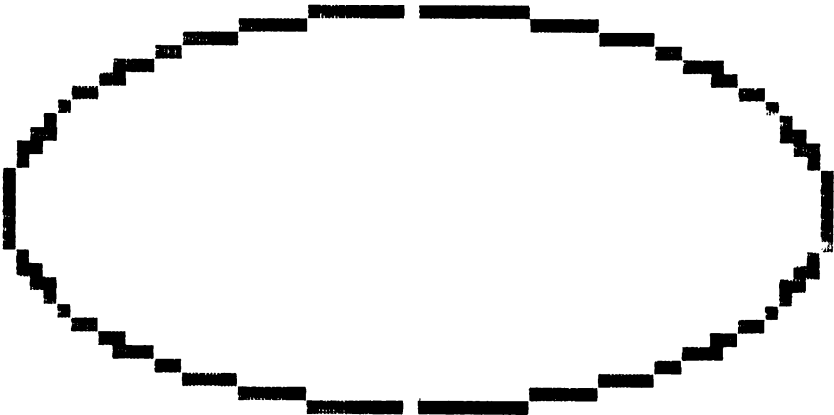
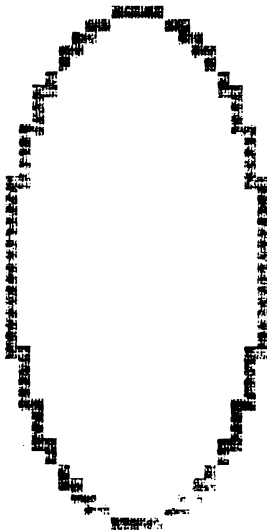



Figura 40

Figura 41

```
  10 REM  DISEGNO DI CILINDRO
  20 REM      TRIDIMENSIONALE
  30 REM  . . . . . . . . . . . . . . . . . . . . . . .
  40 LET X=32
  50 LET Y=32
  60 LET RMA=20
  70 LET RMI=10
  80 LET H=20
  90 GOSUB 6000
 100 STOP
 999 REM ------------------------
```

```
6085 IF C>=180 THEN GOTO 9000
```

```
9000 REM ------------------------
9100 FOR L=B-H TO B
9200 PLOT A,L
9300 NEXT L
9999 GOTO 6090
```

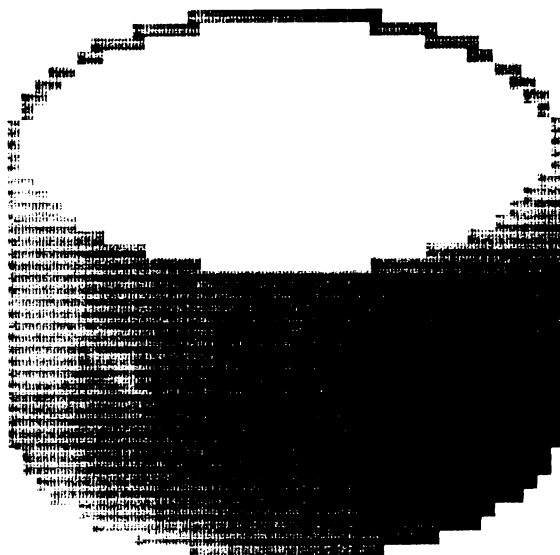



Figura 42

Figura 43

L'esempio fornito per questo programma è particolarmente bello da realizzare, infatti inserendo le istruzioni di fig. 42 nel listato di fig. 38 si ottiene il disegno di un cilindro Tridimensionale (fig. 43), e chi dispone di uno ZX81 (che ha la funzione SLOW) potrà anche vedere il cilindro mentre si forma.
Cilindri di altra forma possono essere disegnati usando altri valori per X, Y, RMA, RMI, H (altezza).

Disegni

I prossimi tre programmi mostrano le capacità del computer nell'eseguire dei disegni. Tali disegni possono essere utili in altri programmi oppure essere usati come esempio dimostrativo fine a se stesso.
Tutti e tre i programmi possono funzionare anche in FAST però in SLOW sono molto meglio perchè si vedono i disegni mentre si formano.

Cielo stellato — FAST O SLOW

Una particolarità del programma (fig. 44) per disegnare un cielo stellato è che il numero delle stelle, piccole e grandi, è maggiore in alto e diminuisce in basso verso l'orizzonte (fig. 45) così come nella realtà.

NEW YORK — FAST O SLOW

Con poche linee di programma (fig. 46) il computer è anche in grado di disegnare in modo casuale il panorama di una città come New York con i suoi grattacieli (fig. 47).

```
  10 REM  CIELO STELLATO
  20 REM -----------------
 100 SLOW
 110 RAND
 120 FOR L=1 TO 22
 130 FOR C=0 TO 31
 140 LET S=1+INT ((9+L*2)*RND)
 150 PRINT AT L-1,C;"█"
 160 IF S=2 THEN PRINT AT L-1,C;
"⊞"
 170 IF S=4 THEN PRINT AT L-1,C;
"▪"
 180 NEXT C
 190 NEXT L
```

Figura 44

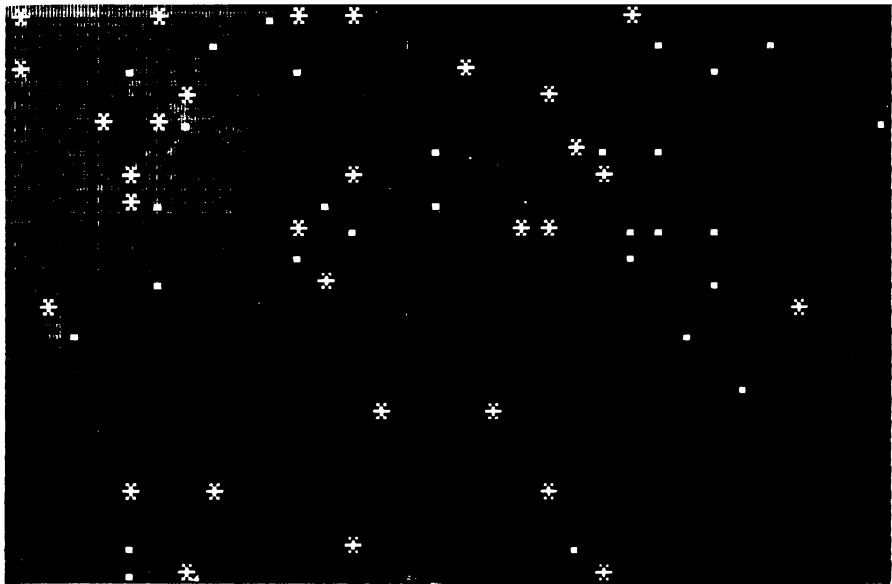



Figura 45

```
  10 REM    NEW  YORK
  20 REM -------------
 100 SLOW
 110 RAND
 120 LET X=0
 130 LET H=2+INT (14*RND)
 140 FOR L=1 TO 2+INT (2*RND)
 150 FOR Y=0 TO H
 160 PLOT X,Y
 165 IF Y>0 AND Y<H AND (4*RND)>
3 THEN UNPLOT X,Y
 170 NEXT Y
 180 LET X=X+1
 190 IF X=64 THEN GOTO 220
 200 NEXT L
 210 GOTO 130
```

Figura 46

Figura 47

Se poi si aggiungono al programma di fig. 46 le linee 77, 88, 99 (fig. 48), le linee 220, 230, 240 (fig. 49) e si sostituiscono le linee 160, 165, con le nuove due linee di fig. 50 si può anche ottenere lo stesso panorama di New York ma di notte (fig. 51).

```
77 FOR P=1 TO 88
88 PRINT "████████████████";
99 NEXT P
```

Figura 48

```
220 FOR T=0 TO 63
230 PLOT T,0
240 NEXT T
```

Figura 49

```
  160 UNPLOT X,Y
  165 IF Y>0 AND Y<H AND (4*RND) >
 3 THEN PLOT X,Y
```

Figura 50

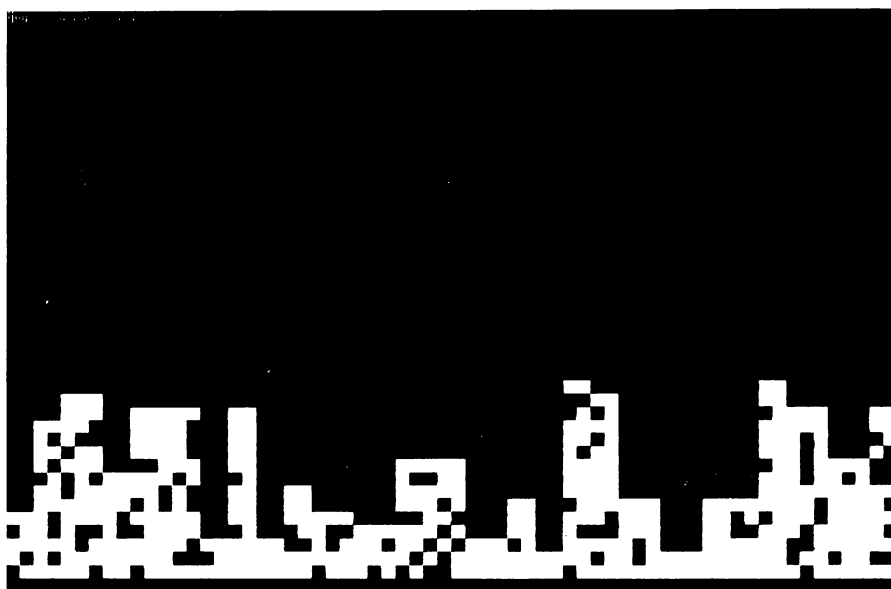



Figura 51

L'ultimo programma grafico (fig. 52) disegna in modo casuale delle montagne (fig. 53) alle quali può essere aggiunto anche un orizzonte inserendo nel programma le linee 270, 280, 290 (fig. 54).

```
  10 REM  MONTAGNE
  20 REM ----------
 100 SLOW
 110 LET A=5
 120 LET M=1
 130 LET X=0
 135 RAND
 140 LET E=1+INT (8*RND)
 150 FOR H=1 TO E
 160 IF M=1 THEN FOR Y=0 TO ABS
(A+H)
 170 IF M=-1 THEN FOR Y=0 TO ABS
(A-H)
 180 IF M=1 AND A+H>=0 OR M=-1 A
ND A-H>=0 THEN PLOT X,Y
 190 NEXT Y
 200 LET X=X+1
 210 IF X=64 THEN GOTO 270
 220 NEXT H
 230 IF M=1 THEN LET A=A+E
 240 IF M=-1 THEN LET A=A-E
 250 LET M=M*(-1)
 260 GOTO 140
```

Figura 52

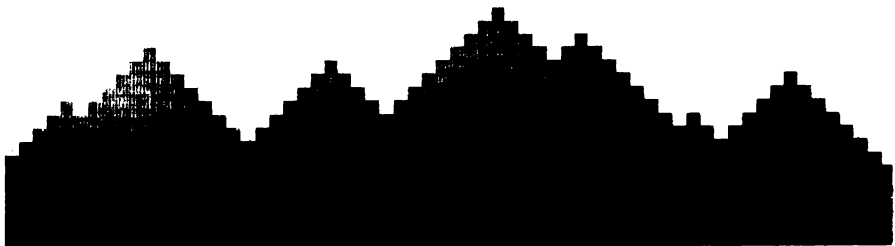



Figura 53

```
270 FOR T=0 TO 31
280 PRINT AT 21,T;"▒"
290 NEXT T
```

Figura 54

È anche possibile, inoltre, aggiungendo le linee 77, 88, 99 (fig. 55) e sostituendo la linea 180 con la nuova linea di fig. 56, ottenere delle montagne chiare di notte.

```
77 FOR P=1 TO 88
88 PRINT "██████████";
99 NEXT P
```

Figura 55

```
 180 IF M=1 AND A+H>=0 OR M=-1 A
ND A-H>=0 THEN UNPLOT X,Y
```

Figura 56

PROGRAMMI VARI

Progettazione di Multivibratori Monostabili con 555

FAST O SLOW

Una delle applicazioni più interessanti del computer è nella progettazione di circuiti elettronici. Un esempio pratico è il programma di fig. 57 che permette di calcolare i valori di un Multivibratore Monostabile con il timer 555 (fig. 58).

```
  10 REM  PROGETTAZIONE DI
  20 REM  MULTIVIBRATORE
  30 REM  MONOSTABILE CON 555
 100 SLOW
 110 PRINT "INSERIRE RA ( 1--100
00 KOHM )"
 115 PRINT "E PREMERE NEWLINE"
 120 INPUT RA
 130 IF RA=0 THEN GOTO 150
 140 IF RA>10000 OR RA<1 THEN GO
TO 120
 150 CLS
 160 PRINT "INSERIRE C ( 0.001--
100 UF )"
 165 PRINT "E PREMERE NEWLINE"
 170 INPUT C
 180 IF C=0 THEN GOTO 200
 190 IF C>100 OR C<0.001 THEN GO
TO 170
 200 CLS
 210 PRINT "INSERIRE T ( 0.01--1
00000 MSEC.)"
 215 PRINT "E PREMERE NEWLINE"
 220 INPUT T
 230 IF T=0 THEN GOTO 250
 240 IF T>100000 OR T<0.01 THEN
GOTO 220
 250 CLS
 260 IF RA=0 AND C=0 AND T=0 THE
N GOTO 110
```

Figura 57 (Continua)

```
 270 IF RA>0 AND C>0 AND T>0 THE
N GOTO 110
 280 IF RA=0 AND C=0 OR RA=0 AND
 T=0 OR C=0 AND T=0 THEN GOTO 11
0
 290 IF RA>0 AND C>0 THEN PRINT
"RA";TAB 6;RA;"  KOHM";TAB 1;TAB
 0;"C";TAB 6;C;"  UF";TAB 1;TAB
0;"T";TAB 6;1.1*RA*C;"  MSEC."
 300 IF RA>0 AND T>0 THEN PRINT
"RA";TAB 6;RA;"  KOHM";TAB 1;TAB
 0;"C";TAB 6;T/(1.1*RA);"  UF";T
AB 1;TAB 0;"T";TAB 6;T;"  MSEC."
 310 IF C>0 AND T>0 THEN PRINT "
RA";TAB 6;T/(1.1*C);"  KOHM";TAB
 1;TAB 0;"C";TAB 6;C;"  UF";TAB
1;TAB 0;"T";TAB 6;T;"  MSEC."
 320 PRINT AT 9,0;"PREMERE NEWLI
NE"
 330 PRINT "PER ALTRI VALORI"
 340 INPUT X$
 350 CLS
 360 GOTO 110
```

Figura 57 (Fine)

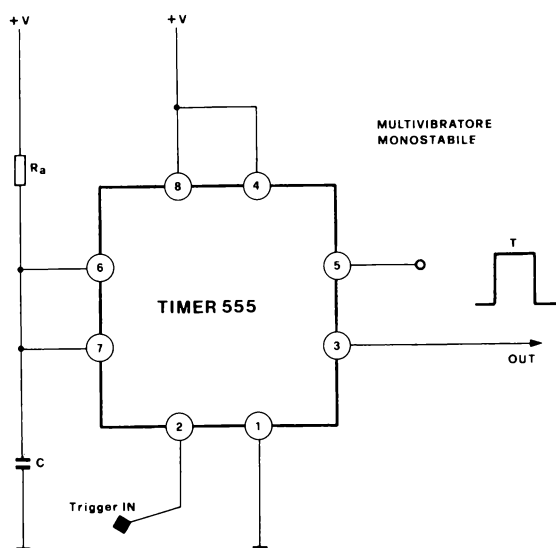



Figura 58

Facendo girare il programma usciranno in sequenza le tre scritte di figg. 59, 60, 61 che chiedono l'inserimento dei tre valori. Se ad esempio si vuole conoscere il valore del tempo T a partire dalla resistenza RA e dal condensatore C si dovrà inserire

```
INSERIRE RA ( 1--10000 KOHM )
E PREMERE NEWLINE
```

Figura 59

```
INSERIRE C ( 0.001--100 UF )
E PREMERE NEWLINE
```

Figura 60

```
INSERIRE T ( 0.01--100000 MSEC.)
E PREMERE NEWLINE
```

Figura 61

```
RA        820    KOHM

C         0.1    UF

T         90.2    MSEC.



PREMERE NEWLINE
PER ALTRI VALORI
```

Figura 62

prima RA e poi C mentre per T (fig. 61) si dovrà inserire il valore 0, in modo simile si dovrà operare se si vuole ricavare RA da C e T o C da RA e T.
Un tipico risultato fornito dal programma è visibile in fig. 62.

Contapezzi **SLOW**

Un uso insolito dello ZX81 può essere quello di contapezzi (fig. 63) capace di contare da 0 a 99.999.999 pezzi.
Facendo girare il programma esce sul video la scritta di fig. 64. Ogni volta che si preme NEWLINE il numero viene incrementato di uno. Oltre a ciò si può inserire anche un numero qualsiasi da aggiungere (per es. 337115) che premendo NEWLI-

```
  10 REM  CONTAPEZZI DA 0
  20 REM  A 99.999.999 PEZZI
 100 SLOW
 110 PRINT "PREMERE NEWLINE PER
1 SOLO PEZZO"
 120 PRINT
 130 PRINT "O INTRODURRE N■ PEZZI
 IN PIU■ O IN"
 140 PRINT
 150 PRINT "MENO (CON -) E PREME
RE NEWLINE"
 160 PRINT AT 10,0;"▛▀▜ ▝ ▛▀▜ ▛▀▀ ▀▀▜ ▀▀▜ ▐
"
 170 PRINT AT 11,0;"▐ ▐ ▛▀▘ ▛▀▘ ▗▘ ▗▘ ▐
"
 180 PRINT AT 12,0;"▘ ▘ ▘ ▀▀▀ ▀▀▀ ▀▀▀ ▘
"
 190 LET N=0
 200 LET C=0
 210 INPUT N$
 220 IF N$="" THEN LET N$="1"
 230 IF CODE N$<>22 THEN FOR A=1
 TO LEN N$
 240 IF CODE N$=22 THEN FOR A=2
TO LEN N$
 250 IF CODE N$(A TO )<28 OR COD
E N$(A TO )>37 THEN GOTO 600
 260 NEXT A
 270 IF (N+VAL N$)>(1E8-1) OR (N
+VAL N$)<0 THEN GOTO 600
 280 LET N=N+VAL N$
 290 DIM B$(1,9-LEN STR$ N)
 300 LET A$=STR$ N+B$(1)
 310 FOR P=1 TO LEN A$-1
 320 LET C=P*2+14
 330 LET J=CODE A$(P TO )
 340 IF J=0 THEN GOSUB 1997
 350 IF J>0 THEN GOSUB (J-28)*10
0+1000
 360 NEXT P
 370 GOTO 210
 600 FOR E=1 TO 3
 610 PRINT AT 15,5;"NUMERO ERRAT
O O ECCESSIVO"
 620 LET T=TAN 2
 630 PRINT AT 15,5;"
"
```

Figura 63 (Continua)

```
 640 LET T=TAN 2
 650 NEXT E
 660 GOTO 210
1000 PRINT AT 10,C;"   "
1010 PRINT AT 11,C;"   "
1020 PRINT AT 12,C;"   "
1030 RETURN
1100 PRINT AT 10,C;"   "
1110 PRINT AT 11,C;"   "
1120 PRINT AT 12,C;"   "
1130 RETURN
1200 PRINT AT 10,C;"   "
1210 PRINT AT 11,C;"   "
1220 PRINT AT 12,C;"   "
1230 RETURN
1300 PRINT AT 10,C;"   "
1310 PRINT AT 11,C;"   "
1320 PRINT AT 12,C;"   "
1330 RETURN
1400 PRINT AT 10,C;"   "
1410 PRINT AT 11,C;"   "
1420 PRINT AT 12,C;"   "
1430 RETURN
1500 PRINT AT 10,C;"   "
1510 PRINT AT 11,C;"   "
1520 PRINT AT 12,C;"   "
1530 RETURN
1600 PRINT AT 10,C;"   "
1610 PRINT AT 11,C;"   "
1620 PRINT AT 12,C;"   "
1630 RETURN
1700 PRINT AT 10,C;"   "
1710 PRINT AT 11,C;"   "
1720 PRINT AT 12,C;"   "
1730 RETURN
1800 PRINT AT 10,C;"   "
1810 PRINT AT 11,C;"   "
1820 PRINT AT 12,C;"   "
1830 RETURN
1900 PRINT AT 10,C;"   "
1910 PRINT AT 11,C;"   "
1920 PRINT AT 12,C;"   "
1930 RETURN
1997 PRINT AT 10,C;"   "
1998 PRINT AT 11,C;"   "
1999 PRINT AT 12,C;"   "
2000 RETURN
```

Figura 63 (Fine)

NE viene sommato al numero presente sullo schermo, oppure si può inserire un numero preceduto dal segno "–" e che premendo NEWLINE viene sottratto dal numero in quel momento presente sullo schermo. Se si cerca di sommare o di sottrarre un numero che farebbe scendere il numero presente sullo schermo sotto lo

```
PREMERE NEWLINE PER 1 SOLO PEZZO
O INTRODURRE N° PEZZI IN PIU' O IN
MENO (CON -) E PREMERE NEWLINE

N°PEZZI

        NUMERO ERRATO O ECCESSIVO
```

Figura 64

```
PREMERE NEWLINE PER 1 SOLO PEZZO
O INTRODURRE N° PEZZI IN PIU' O IN
MENO (CON -) E PREMERE NEWLINE

N°PEZZI    47158302
```

Figura 65

0 o lo farebbe andare oltre il numero massimo di pezzi, il programma segnala l'errore nel modo indicato sempre in fig. 64. Un esempio d'uso dei contapezzi è in fig. 65.
La parte del programma che scrive i numeri di grandi dimensioni può essere utile da inserire anche in altri programmi.
Del contapezzi viene data anche una versione più breve e più rapida da scrivere (fig. 66) che però non usa i numeri grandi ma presenta sul video i numeri di dimensioni normali (fig. 67).

```
  10 REM  CONTAPEZZI DA 0
  20 REM  A 99.999.999 PEZZI
 100 SLOW
 110 PRINT "PREMERE NEWLINE PER
1 SOLO PEZZO"
 120 PRINT
 130 PRINT "O INTRODURRE N■PEZZI
IN PIU■O IN"
 140 PRINT
 150 PRINT "MENO (CON -) E PREME
RE NEWLINE"
 160 PRINT AT 10,2;"N■PEZZI"
 190 LET N=0
 210 INPUT N$
 220 IF N$="" THEN LET N$="1"
 230 IF CODE N$<>22 THEN FOR A=1
 TO LEN N$
 240 IF CODE N$=22 THEN FOR A=2
TO LEN N$
 250 IF CODE N$(A TO )<28 OR COD
E N$(A TO )>37 THEN GOTO 600
 260 NEXT A
 270 IF (N+VAL N$)>(1E8-1) OR (N
+VAL N$)<0 THEN GOTO 600
 280 LET N=N+VAL N$
 290 DIM B$(1,9-LEN STR$ N)
 300 LET A$=STR$ N+B$(1)
 310 PRINT AT 10,12;A$
 320 GOTO 210
 600 FOR E=1 TO 3
 610 PRINT AT 15,0;"INSERITO N■
ERRATO O ECCESSIVO"
 620 LET T=TAN 2
 630 PRINT AT 15,0;"
                   "
 640 LET T=TAN 2
 650 NEXT E
 660 GOTO 210
```

Figura 66

```
PREMERE NEWLINE PER 1 SOLO PEZZO

O INTRODURRE N■PEZZI IN PIU■O IN

MENO (CON -) E PREMERE NEWLINE

   N■PEZZI   87654321
```

Figura 67

Una delle applicazioni del programma di "Scritte rotanti" (fig. 68) può essere nella pubblicità e nell'informazione in genere.

```
  10 REM   SCRITTE ROTANTI
 100 SLOW
 110 PRINT "INSERIRE FRASE"
 120 INPUT A$
 130 CLS
 140 PRINT "INSERIRE VELOCITA' (
1--10 )"
 150 INPUT T
 160 CLS
 170 PRINT AT 3,0;"████████████████
████████████████"
 180 PRINT AT 7,0;"████████████████
████████████████"
 190 IF LEN A$<32 THEN DIM H$(32
-LEN A$+10)
 195 IF LEN A$>31 THEN DIM H$(10
)
 200 LET B$=A$+H$
 210 LET C$=B$
 220 PRINT AT 5,0;C$(1 TO 32)
 230 LET C$=C$(2 TO )
 240 IF LEN C$<32 THEN LET C$=C$
+B$
 250 FOR D=1 TO T
 260 NEXT D
 270 GOTO 220
```

Figura 68

```
████████████████████████████████

   QUESTA E' UNA SCRITTA ROTANTE .

████████████████████████████████
```

Figura 69

```
████████████████████████████████

NA SCRITTA ROTANTE.  QUESTA E' UN

████████████████████████████████
```

Figura 70

Dopo avere premuto RUN e NEWLINE si inserisce prima la frase anche di molti caratteri e poi la velocità di rotazione. Dopodichè i caratteri, i numeri e i simboli della frase inserita iniziano a ruotare da destra a sinistra.
Le figure 69 e 70, 71 e 72, 73 e 74 mostrano tre esempi di scritte rotanti ciascuno dei

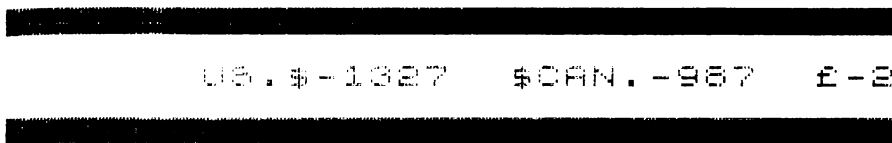



Figura 71

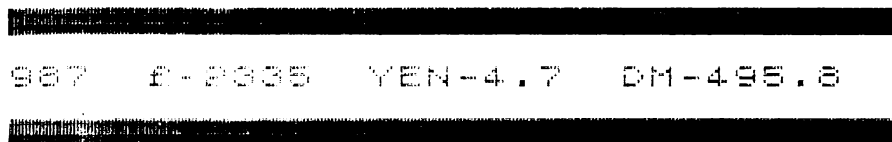



Figura 72

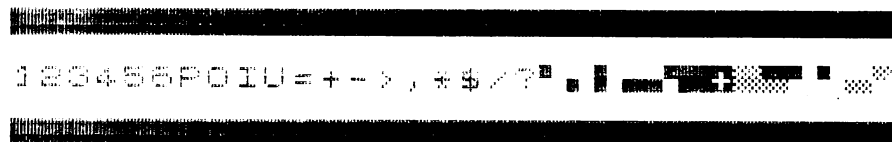



Figura 73

Figura 74

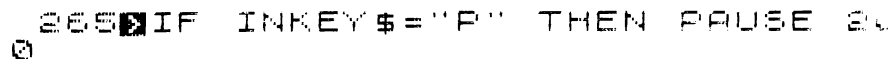



Figura 75

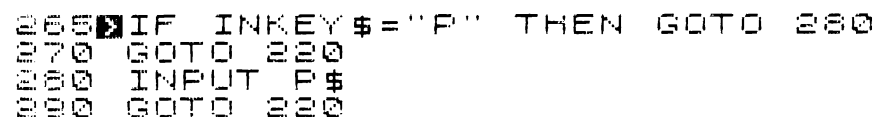



Figura 76

quali visto in due differenti momenti, comunque il modo migliore per rendersi conto del funzionamento è quello di provare il programma in pratica.
Aggiungendo la linea 265 (fig. 75) si può rallentare la rotazione della frase di 4 secondi ad ogni carattere (PAUSE 200) se si tiene premuto il tasto “P”.

Figura 77

```
201 REM  LINEE 204>>>209
202 REM  ROUTINE INVERSIONE
203 REM  CARATTERI DELLA FRASE
204 DIM I$(1,LEN B$)
205 FOR I=1 TO LEN B$
206 IF CODE B$(I TO I)<64 THEN
LET I$(1,I)=CHR$ (CODE B$(I TO I
)+128)
 207 IF CODE B$(I TO I)>127 THEN
 LET I$(1,I)=CHR$ (CODE B$(I TO
I))
 208 NEXT I
 209 LET B$=I$(1)
```

Figura 78

```
A$= 1234567890POIUYTREWQASDFGHJK
L.MNBVCXZ,><*/?;:=+-)($$$£$$$$
```

Figura 79

```
67890POIUYTREWQASDFGHJKL.MNBVCXZ
```

Figura 80

È anche possibile, inserendo le linee 265, 280 e 290 di fig. 76, fare in modo di bloccare la rotazione della frase ogni volta che si preme il tasto “P” e fino a che non si preme NEWLINE.
Un’altra possibilità ancora è quella di avere una frase in caratteri inversi anche inserendo dei caratteri normali, per ottenere ciò occorre modificare le linee 170 e 180 nel modo indicato in fig. 77 ed aggiungere al programma le nove linee di fig. 78, dopodichè inserendo, per esempio, la frase di fig. 79 si ottiene una scritta rotante come quella di fig. 80.

Un programma di Renumber è molto utile durante la programmazione poichè permette di riordinare automaticamente tutti i numeri di linea.
La versione di fig. 81 realizza questo riordino esclusi i numeri dopo GOTO e

```
9000 STOP
9010 REM      RENUMBER
9020 FAST
9025 PRINT "INSERIRE N■LINEA INI
ZIO RENUMBER"
9030 INPUT NIR
9032 CLS
9035 PRINT "INSERIRE NUOVO N■LIN
EA INIZIALE"
9040 INPUT NLI
9042 CLS
9045 PRINT "INSERIRE DISTANZA TR
A LINEE"
9050 INPUT DTL
9052 CLS
9055 LET IM=16508
9060 LET NX=PEEK (IM+1)*256+PEEK
 (IM+2)
9065 IF NX<NIR THEN GOTO 9110
9070 IF NX=9000 THEN GOTO 9150
9080 POKE (IM+1),INT (NLI/256)
9090 POKE (IM+2),NLI-256*INT (NL
I/256)
9100 LET NLI=NLI+DTL
9110 LET IM=IM+5
9120 IF PEEK IM=118 THEN GOTO 90
60
9130 LET IM=IM+1
9140 GOTO 9120
9150 LIST
```

Figura 81

```
INSERIRE NUOVO N■LINEA INIZIALE
```

Figura 82

```
INSERIRE N■LINEA INIZIO RENUMBER
```

Figura 83

```
INSERIRE DISTANZA TRA LINEE
```

Figura 84

```
   1 REM  PROGRAMMA PROVA
   2 REM  PER RENUMBER
   3 REM  LINEE 1---7777
   4 REM -----------------
   8 LET A=2
  18 REM .
  22 REM .
  77 REM .
 115 GOTO 1007
 235 REM .
 532 REM .
 550 REM .
 666 REM .
 890 REM .
 995 GOSUB 4
1003 REM ....XX
1235 REM .
2222 REM .
3578 REM .
4521 REM .
5744 REM K
7777 REM  FINE PROGRAMMA PROVA
```

Figura 85

```
 1 REM  PROGRAMMA PROVA
 2 REM  PER RENUMBER
 3 REM  LINEE 1---7777
 4 REM -----------------
 5 LET A=2
 6 REM .
 7 REM .
 8 REM .
 9 GOTO 1007
10 REM .
11 REM .
12 REM .
13 REM .
14 REM .
15 GOSUB 4
16 REM ....XX
17 REM .
18 REM .
19 REM .
20 REM .
21 REM K
22 REM  FINE PROGRAMMA PROVA
```

Figura 86

```
100 REM  PROGRAMMA PROVA
110 REM  PER RENUMBER
120 REM  LINEE 1---7777
130 REM ----------------------
140 LET A=2
150 REM .
160 REM .
170 REM .
180 GOTO 1007
190 REM .
200 REM .
210 REM .
220 REM .
230 REM .
240 GOSUB 4
250 REM ....XX
260 REM .
270 REM .
280 REM .
290 REM .
300 REM K
310 REM  FINE PROGRAMMA PROVA
```

Figura 87

GOSUB che vanno modificati singolarmente, in tal modo si ha un programma di Renumber più facile da inserire e togliere quando occorre.
Il programma va fatto girare con un RUN 9Ø1Ø o con un GOTO 9Ø1Ø. Dopo di ciò si vedranno apparire sul video per prima la scritta di fig. 82 che chiede di inserire il numero da cui iniziare la rinumerazione (per es. 3), poi la scritta di fig. 83 che chiede di inserire il nuovo numero di linea iniziale (per es. 100) e infine la scritta di fig. 84 che chiede di inserire la distanza desiderata tra le linee (per es. 10).
La fig. 85 mostra come esempio un programma che ha proprio bisogno di essere riordinato, mentre le figure 86 e 87 mostrano due differenti esempi di rinumerazione del programma di fig. 85 ottenute con il programma di Renumber appena descritto.

Lettore Memoria

FAST O SLOW

Un'altra utility interessante è il lettore della memoria del computer (fig. 88) con il quale si può leggere il contenuto di una parte qualsiasi della ROM o della RAM a partire dall'indirizzo introdotto all'inizio (fig. 89).
La fig. 90 mostra un esempio di lettura della RAM ed in particolare proprio di una zona nella quale è scritto il programma di fig. 88.

```
  10 REM  LETTORE MEMORIA
 100 SLOW
 110 PRINT "INSERIRE INDIRIZZO D
I PARTENZA"
 120 PRINT "( 0--65535 ) E PREME
RE NEWLINE"
 130 INPUT A
 140 IF A<0 OR A>65535 THEN GOTO
 130
 150 CLS
 160 PRINT " INDIRIZZO  CONTENUT
O CARATTERE"
 170 PRINT "--------------------
------------"
 180 FOR L=2 TO 21
 190 PRINT AT L,3;A+L-2;TAB 15;P
EEK (A+L-2)
 200 IF PEEK (A+L-2)<>118 THEN P
RINT AT L,24;CHR$ PEEK (A+L-2)
 210 NEXT L
 220 COPY
 230 INPUT X$
 240 CLS
 250 GOTO 110
```

Figura 88

```
INSERIRE INDIRIZZO DI PARTENZA
( 0--65535 ) E PREMERE NEWLINE
```

Figura 89

INDIRIZZO	CONTENUTO	CARATTERE
16513	234	REM
16514	0	
16515	49	L
16516	42	E
16517	57	T
16518	57	T
16519	52	O
16520	55	R
16521	42	E
16522	0	
16523	50	M
16524	42	E
16525	50	M
16526	52	O
16527	55	R
16528	46	I
16529	38	A
16530	118	
16531	0	
16532	100	?

Figura 90

Il programma ha un'istruzione (220) che stampa il pezzo di memoria appena esaminato, dopodichè premendo NEWLINE si può introdurre un nuovo indirizzo di partenza.

Programma caricamento routines in linguaggio macchina contenute in stringa

FAST

Se non si hanno limitazioni di memoria il modo più chiaro e più comodo per inserire in un programma delle routines in linguaggio macchina è quello di caricare le varie istruzioni decimali (con uno spazio tra l'una e l'altra) in una stringa (per es. A$) e di utilizzare il programma di fig. 91 (istruzioni da 2000 a 2090), programma utilizzato in pratica anche in due delle animazioni del prossimo capitolo.
Le linee da 1 a 80 in fig. 91 servono per mostrare il funzionamento della routine in linguaggio macchina contenuta in A$ (linea 2020).

```
   1 REM  PROGRAMMA DI PROVA
  10 GOSUB 2000
  20 PRINT "INSERIRE N ( 1--255
) "
  30 INPUT N
  35 CLS
  40 POKE 16507,N
  50 PRINT USR 20225
  60 FAST
  70 STOP
1000 REM  ROUTINE CARICAMENTO
1001 REM  PROGRAMMI IN
1002 REM  LINGUAGGIO MACCHINA
1003 REM  A PARTIRE DA
1004 REM  ISTRUZIONI DECIMALI
2000 FAST
2020 LET A$="042 123 064 041 041
 068 077 201"
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 4
2050 POKE A,VAL A$(L TO L+2)
2060 LET A=A+1
2070 NEXT L
2080 SLOW
2090 RETURN
```

Figura 91

```
20225    042         LD HL,(16507)
20226    123
20227    064
20228    041         ADD HL,HL
20229    041         ADD HL,HL
20230    068         LD B,H
20231    077         LD C,L
20232    201         RETURN
```

Figura 92

```
INSERIRE N ( 1--255 )
```

Figura 93

```
2015 REM  MODIFICA --1--
2020 LET A$="0421230640410410680
77201"
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 3
2050 POKE A,VAL A$(L TO L+2)
2060 LET A=A+1
2070 NEXT L
```

Figura 94

```
2015 REM  MODIFICA --2--
2020 INPUT A$
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 3
```

Figura 95

Tale routine di esempio (fig. 92) serve solo a moltiplicare per quattro un numero inserito all'inizio del programma (fig. 93) e a presentare il risultato sul video. Modificando alcune linee del programma nel modo indicato in fig. 94, si possono eliminare gli spazi tra le istruzioni contenute nella stringa (linea 2020) ogni istruzione, però, deve essere sempre formata da tre numeri anche quando è inferiore a 100 (per es. 2 va scritto 002).

È anche possibile sostituire la stringa con le istruzioni della linea 2020 con un INPUT A$ (fig. 95) in modo da potere inserire ogni volta delle routines da provare. Oltre che in decimali è possibile caricare le istruzioni nella stringa anche in esadecimale utilizzando il programma di fig. 96 e le istruzioni esadecimali di esempio contenute in A$ sono sempre quelle di fig. 92. Un ulteriore compattazione della stringa può essere ottenuta eliminando gli spazi tra un'istruzione e l'altra, utilizzando l'ultima modifica esposta in fig. 97.
Per convertire delle istruzioni decimali in esadecimali, e viceversa, si possono usare gli appositi programmi per conversioni di numeri in base diversa che sono presentati più avanti in questo stesso capitolo.

```
1000 REM  ROUTINE CARICAMENTO
1001 REM  PROGRAMMI IN
1002 REM  LINGUAGGIO MACCHINA
1003 REM  A PARTIRE DA
1004 REM  ISTRUZIONI ESADECIMALI
1005 REM  MEMORIZZATE IN A$
2000 FAST
2020 LET A$="2A 7B 40 29 29 44 4
D C9"
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 3
2050 POKE A,(CODE A$(L TO )-28)*
16+(CODE A$(L+1 TO )-28)
2060 LET A=A+1
2070 NEXT L
```

Figura 96

```
2015 REM  MODIFICA --3--
2020 LET A$="2A7B40292944 4DC9"
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 2
```

Figura 97

Programma caricamento routines in linguaggio macchina contenute in 1 REM — FAST

Un sistema già noto per inserire delle istruzioni in linguaggio macchina in un programma è quello di scrivere tali istruzioni (in esadecimale) nell'istruzione 1 REM posta all'inizio del programma.
Rispetto a programmi simili già esistenti la versione di fig. 98 (linee da 2000 a 2090 più linee 1 e 2) permette di inserire nella istruzione 1 REM un numero qualsiasi di

```
   1 REM 2A7640292944 4DC9
   2 GOSUB 2000
  20 PRINT "INSERIRE N ( 1--255
) "
  30 INPUT N
  35 CLS
  40 POKE 16507,N
  50 PRINT USR 20225
  60 FAST
  80 STOP
1000 REM  ROUTINE CARICAMENTO
1001 REM  PROGRAMMI IN
1002 REM  LINGUAGGIO MACCHINA
1003 REM  A PARTIRE DA
1004 REM  ISTRUZIONI ESADECIMALI
1005 REM  MEMORIZZATE IN <1 REM>
2000 FAST
2020 LET S=16511+PEEK 16511+256*
PEEK 16512
2030 LET A=20225
2040 FOR L=16514 TO S STEP 2
2050 POKE A,((PEEK L)-28)*16+((P
EEK (L+1))-28)
2060 LET A=A+1
2070 NEXT L
2080 SLOW
2090 RETURN
```

Figura 98

istruzioni esadecimali senza modificare il programma il quale controlla automaticamente la lunghezza del contenuto di 1 REM con l'istruzione contenuta nella linea 2020.
Se la routine in linguaggio macchina è più lunga di 200-220 Bytes o se si usa la RAM da 16K l'istruzione 2030 va modificata in conseguenza.

Conversioni numeriche

I prossimi sei programmi eseguono le conversioni tra i tre principali tipi di numerazione utilizzati dai programmatori, le numerazioni binaria, decimale e esadecimale. Di ciascun programma viene dato anche un esempio pratico.
Le applicazioni dei programmi di conversione possono essere varie, per esempio la conversione da decimale a esadecimale può essere utile per convertire una serie di

istruzioni decimali in linguaggio macchina nei corrispondenti valori esadecimali da inserire in un programma.
Tenete presente che la matematica del computer ha una precisione di 8-9 Digit per cui se il numero binario, decimale o esadecimale inserito è troppo grande il numero ottenuto dalla conversione può risultare arrotondato.
In tutti e sei i programmi i numeri da convertire vanno inseriti dal Bit più significativo (MSB) al Bit meno significativo (LSB).
Tutti i programmi hanno un'istruzione finale per la stampa del risultato.

Conversione da binario a decimale **FAST O SLOW**

Programma in fig. 99. Esempio in fig. 100

```
  10 INPUT A$
1000 REM  CONVERSIONE DA
1010 REM  BINARIO A DECIMALE
1020 FAST
1030 LET N=0
1040 LET K=0
1050 FOR C=LEN A$-1 TO 0 STEP -1
1060 LET K=K+1
1070 LET N=N+2**C*VAL A$(K TO K)
1080 NEXT C
1090 CLS
1100 PRINT "N■ BINARIO:"
1110 PRINT A$
1120 PRINT
1130 PRINT "= DECIMALE      ";N
1140 COPY
```

Figura 99

```
N■ BINARIO:
1100011011011101

= DECIMALE       50909
```

Figura 100

Conversione da binario a esadecimale **FAST O SLOW**

Programma in fig. 101. Esempio in fig. 102

```
  10 INPUT A$
1500 REM  CONVERSIONE DA
1510 REM  BINARIO A ESADECIMALE
1520 FAST
1530 LET S=0
1540 LET L=LEN A$/4-INT (LEN A$/
4)
1550 IF L=0 THEN GOTO 1620
1560 LET S=1
1570 IF L<0.26 THEN LET S=S+2
1580 IF L>0.26 AND L<0.51 THEN L
ET S=S+1
1590 DIM S$(1,S)
1600 LET S$(1)="000"
1610 LET A$=S$(1)+A$
1620 LET H=LEN A$/4
1630 DIM N(H)
1640 FOR F=1 TO H
1650 LET N(F)=0
1660 FOR C=0 TO 3
1670 LET N(F)=N(F)+2**C*VAL A$(F
*4-C TO F*4-C)
1680 NEXT C
1690 NEXT F
1700 CLS
1710 PRINT "N■ BINARIO:"
1720 IF S=0 THEN PRINT A$
1730 IF S>0 THEN PRINT A$(S+1 TO
 )
1740 PRINT
1750 PRINT "= ESADECIMALE    ";
1760 FOR P=1 TO H
1770 PRINT CHR$ (N(P)+28);
1780 NEXT P
1790 PRINT
1800 COPY
```

Figura 101

```
N■ BINARIO:
1100011011011101

= ESADECIMALE    C6DD
```

Figura 102

Conversione da esadecimale a decimale

FAST O SLOW

Programma in fig. 103. Esempio in fig. 104

```
  10 INPUT A$
2000 REM  CONVERSIONE DA
2010 REM  ESADECIMALE 9 DECIMALE
2020 FAST
2030 LET N=0
2040 FOR J=0 TO LEN A$-1
2050 LET N=N+((CODE A$(LEN A$-J
TO )-28)*16**J)
2060 NEXT J
2070 CLS
2080 PRINT "N■ ESADECIMALE:      "
;A$
2090 PRINT
2100 PRINT "= DECIMALE      ";N
2110 COPY
```

Figura 103

```
N■ ESADECIMALE:      C6DD

= DECIMALE      50909
```

Figura 104

Conversione da esadecimale a binario

FAST O SLOW

Programma in fig. 105. Esempio in fig. 106

```
  10 INPUT A$
2500 REM  CONVERSIONE DA
2510 REM  ESADECIMALE A BINARIO
2520 FAST
2530 DIM N$(1,LEN A$*4)
2540 LET B$="0000 0001 0010 0011
 0100 0101 0110 0111 1000 1001 1
010 1011 1100 1101 1110 1111"
2550 FOR E=1 TO LEN A$
2560 LET C=CODE A$(E TO )-27
2570 LET N$(1,E*4-3 TO E*4)=B$(C
*5-4 TO C*5-1)
2580 NEXT E
2590 FOR Z=1 TO LEN N$(1)
2600 LET R$=N$(1,Z TO )
```

Figura 105 (Continua)

```
2610 IF N$(1,Z)="1" THEN GOTO 26
30
2620 NEXT Z
2630 CLS
2640 PRINT "N■ ESADECIMALE:
;A$
2650 PRINT
2660 PRINT "= BINARIO  ( MSB >>>
> LSB )"
2670 PRINT R$
2680 COPY
```

Figura 105 (Fine)

```
N■  ESADECIMALE:    C6DD

= BINARIO  ( MSB >>>> LSB )
1100011011011101
```

Figura 106

Conversione da decimale a esadecimale **FAST O SLOW**

Programma in fig. 107. Esempio in fig. 108

```
  10 INPUT N
3000 REM  CONVERSIONE DA
3010 REM  DECIMALE A ESADECIMALE
3020 FAST
3030 LET E=0
3040 IF 15*16**E>N THEN GOTO 307
0
3050 LET E=E+1
3060 GOTO 3040
3070 LET H$=""
3080 LET J=N
3090 FOR C=E TO 0 STEP -1
3100 FOR F=15 TO 0 STEP -1
3110 LET K=F*16**C
3120 IF J-K>=0 THEN LET H$=H$+CH
R$ (28+F)
3130 IF J-K>=0 THEN GOTO 3150
3140 NEXT F
```

Figura 107 (Continua)

```
3150 LET J=J-K
3160 NEXT C
3170 IF LEN H$>1 AND CODE H$=28
THEN LET H$=H$(2 TO )
3180 CLS
3190 PRINT "N■ DECIMALE:      ";N
3200 PRINT
3210 PRINT "= ESADECIMALE      ";H
$
3220 COPY
```

Figura 107 (Fine)

```
N■  DECIMALE:      50909

= ESADECIMALE       C6DD
```

Figura 108

Conversione da decimale a binario **FAST O SLOW**

Programma in fig. 109. Esempio in fig. 110

```
  10 INPUT N
3500 REM  CONVERSIONE DA
3510 REM  DECIMALE A BINARIO
3520 FAST
3530 LET E=0
3540 IF 2**E>N THEN GOTO 3570
3550 LET E=E+1
3560 GOTO 3540
3570 LET B$=""
3580 LET J=N
3590 FOR C=E-1 TO 0 STEP -1
3600 LET K=2**C
3610 IF J-K>=0 THEN LET B$=B$+"1
"
3620 IF J-K<0 THEN LET B$=B$+"0"
3630 IF J-K>=0 THEN LET J=J-K
3640 NEXT C
3650 CLS
3660 PRINT "N■ DECIMALE:      ";N
3670 PRINT
3680 PRINT "= BINARIO  (MSB >>>
> LSB)"
3690 PRINT B$
3700 COPY
```

Figura 109

```
N° DECIMALE:      50909

= BINARIO  ( MSB >>>> LSB )
1100011011011101
```

Figura 110

Programma di riunione delle routines di conversione numerica SLOW

I sei programmi di conversione numerica appena descritti possono essere riuniti in una sorta di calcolatrice per Programmatori togliendo da tutti e sei i programmi le linee 10 ed aggiungendoli al programma di fig. 111.

```
  10 REM  CALCOLATRICE
  20 REM  PER PROGRAMMATORI
 100 SLOW
 110 PRINT "PREMERE.        PER INSE
RIRE N° "
 120 PRINT "--------------------------------
-----------"
 130 PRINT TAB 5;"B          BINARI
O"
 140 PRINT TAB 5;"D          DECIMA
LE"
 150 PRINT TAB 5;"E          ESADEC
IMALE"
 160 IF INKEY$="B" THEN GOTO 200
 170 IF INKEY$="D" THEN GOTO 350
 180 IF INKEY$="E" THEN GOTO 500
 190 GOTO 160
 200 CLS
 210 PRINT "INSERIRE N° IN BINAR
IO ( 1/0 )"
 220 PRINT "( MSB >>> LSB )E PRE
MERE NEWLINE"
 230 INPUT A$
 240 FOR A=1 TO LEN A$
 250 IF CODE A$(A TO )>29 OR COD
E A$(A TO )<28 THEN GOTO 230
 260 NEXT A
 270 CLS
 280 PRINT "PREMERE:       PER CONV
ERTIRE IN"
 290 PRINT "--------------------------------
-----------"
```

Figura 111 (Continua)

```
 300 PRINT TAB 5;"D      DECIMAL
E"
 310 PRINT TAB 5;"E      ESADECI
MALE"
 320 IF INKEY$="D" THEN GOTO 100
0
 330 IF INKEY$="E" THEN GOTO 150
0
 340 GOTO 320
 350 CLS
 360 PRINT "INSERIRE N■ DECIMALE
"
 370 PRINT "E PREMERE NEWLINE"
 380 INPUT A$
 390 FOR A=1 TO LEN A$
 400 IF CODE A$(A TO )>37 OR COD
E A$(A TO )<28 THEN GOTO 380
 410 NEXT A
 415 LET N=VAL A$
 420 CLS
 430 PRINT "PREMERE:     PER CONV
ERTIRE IN"
 440 PRINT "---------------------
-----------"
 450 PRINT TAB 5;"B      BINARIO
"
 460 PRINT TAB 5;"E      ESADECI
MALE"
 470 IF INKEY$="B" THEN GOTO 350
0
 480 IF INKEY$="E" THEN GOTO 300
0
 490 GOTO 470
 500 CLS
 510 PRINT "INSERIRE N■ ESADECIM
ALE"
 520 PRINT "( 0 >> F ) E PREMERE
 NEWLINE"
 530 INPUT A$
 540 FOR A=1 TO LEN A$
 550 IF CODE A$(A TO )>43 OR COD
E A$(A TO )<28 THEN GOTO 530
 560 NEXT A
 570 CLS
 580 PRINT "PREMERE:     PER CONV
ERTIRE IN"
 590 PRINT "---------------------
-----------"
```

Figura 111 (Continua)

```
 600 PRINT TAB 5;"B          BINARIO
"
 610 PRINT TAB 5;"D          DECIMAL
E"
 620 IF INKEY$="B" THEN GOTO 250
0
 630 IF INKEY$="D" THEN GOTO 200
0
 640 GOTO 620
 650 PRINT AT 10,0;"PREMERE NEWL
INE PER"
 660 PRINT AT 11,0;"UN■ALTRA CON
VERSIONE"
 670 INPUT X$
 680 CLS
 690 GOTO 100
1150 GOTO 650
1810 GOTO 650
2120 GOTO 650
2690 GOTO 650
3230 GOTO 650
3710 GOTO 650
```

Figura 111 (Fine)

```
PREMERE:      PER INSERIRE N■
-----------------------------
       B          BINARIO
       D          DECIMALE
       E          ESADECIMALE
```

Figura 112

```
INSERIRE N■ IN BINARIO ( 1/0 )
( MSB >>> LSB )E PREMERE NEWLINE
```

Figura 113

```
INSERIRE N■ DECIMALE
E PREMERE NEWLINE
```

Figura 114

```
INSERIRE N■ ESADECIMALE
( 0 >> F ) E PREMERE NEWLINE
```

Figura 115

Dopo di ciò, dando il RUN, il programma chiede all'inizio che tipo di numero si vuole convertire (fig. 112).
A seconda del tasto premuto uscirà sul video una delle tre richieste di inserimento di un numero visibile nelle figure 113, 114 e 115 a cui seguirà una delle corrispondenti tabelle delle figure 116, 117 e 118 che chiede in quale codice deve essere convertito il numero inserito.

```
PREMERE:        PER CONVERTIRE IN
---------------------------------
      D         DECIMALE
      E         ESADECIMALE
```

Figura 116

```
PREMERE:        PER CONVERTIRE IN
---------------------------------
      B         BINARIO
      E         ESADECIMALE
```

Figura 117

```
PREMERE:        PER CONVERTIRE IN
---------------------------------
      B         BINARIO
      D         DECIMALE
```

Figura 118

```
PREMERE NEWLINE PER
UN' ALTRA CONVERSIONE
```

Figura 119

A seconda della conversione effettuata il risultato sul video sarà uno di quelli già visti come esempio nei sei programmi precedenti (figure 100, 102, 104, 106, 108, 110) insieme alla scritta di fig. 119 che chiede se si vuole effettuare un'altra conversione.

Nelle conversioni da decimale a binario e da decimale a esadecimale le cifre 67.092.480 e 67.108.864 sono rispettivamente i due numeri massimi da inserire oltre i quali il programma può sbagliare o arrotondare.

ANIMAZIONI

Countdown e lancio missile **SLOW**

Il primo programma di animazione (fig. 120) disegna sullo schermo un cielo notturno con una rampa di lancio ed un missile pronto a partire (fig. 121). Terminato il conteggio alla rovescia che dura 10 secondi il missile parte (fig. 122) per poi sparire nella parte alta dello schermo.
Per ottenere un movimento rapido del missile e per non vedere il disegno dello stesso formarsi ad ogni spostamento è stato necessario utilizzare una breve routine in linguaggio macchina contenuta nella stringa A$ (linea 130) e caricata nella RAM all'inizio del programma. La routine viene descritta più dettagliatamente quì di seguito:

Indirizzo RAM	Codice decimale	Mnemonico
20224	001	LD BC, nn
20225	000	Numero caricato dal BASIC
20226	000	
20227	017	
20228	033	LD DE, 33
20229	000	
20330	042	
20331	012	LD HL, (16396)
20332	064	
20333	009	ADD, HL, BC
20334	054	LD (HL), 5
20335	005	
20336	025	ADD HL, DE
20337	054	LD (HL), 5
20338	005	
20339	025	ADD HL, DE
20340	054	LD (HL), 6
20341	006	
20342	035	INC HL
20343	054	LD (HL), 129
20344	129	
20345	043	DEC HL
20346	025	ADD HL, DE
20347	054	LD (HL), 128
20348	128	
20349	035	INC HL
20350	054	LD (HL), 128
20351	128	
20352	201	RETURN

```
  10 REM   COUNTDOWN
  20 REM   E LANCIO MISSILE
  30 REM   ----------------
 100 FAST
 120 LET A=20224
 130 LET A$="001 000 000 017 033
000 042 012 064 009 054 005 025
054 005 025 054 006 035 054 129
043 025 054 128 035 054 128 201
"
 140 FOR J=1 TO LEN A$ STEP 4
 150 POKE A,VAL A$(J TO J+2)
 160 LET A=A+1
 170 NEXT J
 180 FOR F=1 TO 88
 190 PRINT "████████";
 200 NEXT F
 210 PRINT AT 18,20;"X"
 220 PRINT AT 19,20;"XX"
 230 PRINT AT 20,20;"XX"
 240 PRINT AT 21,0;"████████████
████████████████████"
 250 PRINT AT 18,18;"█ █"
 260 PRINT AT 19,18;"█ █"
 270 PRINT AT 20,18;"▗▟█"
 280 RAND
 290 FOR S=1 TO 20
 300 LET L=1+INT (RND*14)
 310 LET C=1+INT (RND*31)
 320 IF C>15 AND C<21 THEN GOTO
310
 330 LET K=1+INT (RND*2)
 340 IF K=1 THEN PRINT AT L,C;"█
"
 350 IF K=2 THEN PRINT AT L,C;"█
"
 360 NEXT S
 370 SLOW
 380 PRINT AT 19,26;"00:0"
 390 FOR T=9 TO 0 STEP -1
 400 FOR D=1 TO 25
 410 NEXT D
 420 PRINT AT 19,30;T
 430 NEXT T
 440 FOR H=580 TO 19 STEP -33
 450 POKE 20225,H-(INT (H/256)*2
56)
```

Figura 120 (Continua)

```
460 POKE 20226,INT (H/256)
470 LET Z=USR 20224
480 NEXT H
490 PRINT AT 0,18;"█"
500 PRINT AT 1,18;"█"
510 PRINT AT 2,18;"██"
```

Figura 120 (Fine)

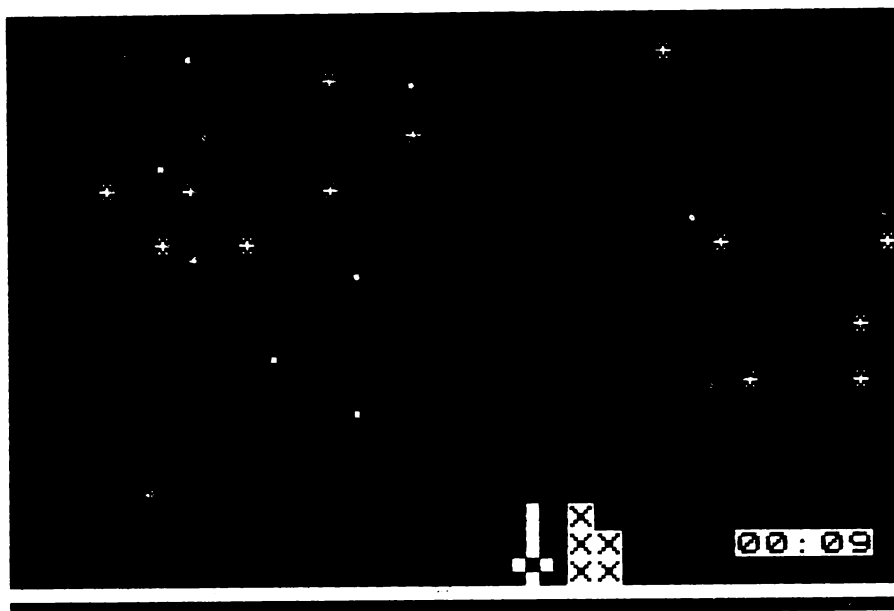



Figura 121

CaLeIdOsCoPiO SLOW

In un caleidoscopio diviso in 4 zone un oggetto visibile in una zona deve apparire immediatamente e simmetricamente nelle altre tre.
Per questo scopo la grafica del BASIC non è sufficientemente veloce, per cui anche in questo secondo programma di animazione (fig. 123) la maggiore velocità nel disegnare sullo schermo è ottenuta con la routine in linguaggio macchina contenuta nella stringa A$ (linea 300).
La fig. 124 riporta un esempio di disegno prodotto dal programma, disegno che si modifica continuamente e velocemente sullo schermo.

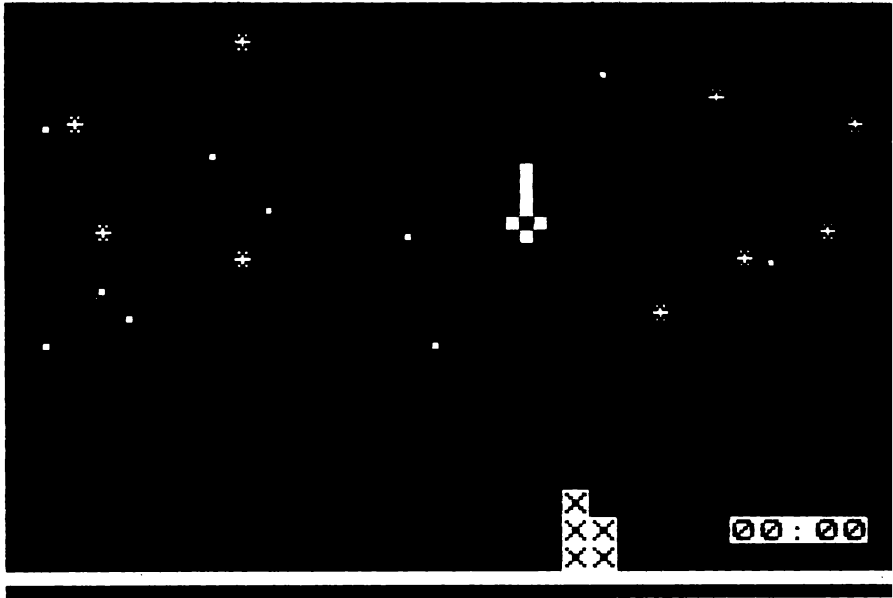



Figura 122

Così com'è il programma disegna sullo schermo in media un terzo di caratteri normali, un terzo di caratteri inversi ed un terzo di spazi bianchi, modificando le linee 180, 190, 200 come in fig. 125, il programma disegnerà sullo schermo solo caratteri grafici, diretti e inversi e spazi bianchi (fig. 126), che cambiano di continuo.
Il programma può essere fermato premendo il tasto S.
Qui di seguito viene esposta più in dettaglio la routine in linguaggio macchina utilizzata dal programma:

Indirizzo RAM	Codice decimale	Mnemonico
20230	022	LD D, n (n=X)
20231	000	
20232	042	LD HL, (16396)
20233	012	
20234	064	
20235	237	LD BC, (20225)
20236	075	
20237	001	
20238	079	
20239	009	ADD HL, BC
20240	237	LD BC, (20227)
20241	075	
20242	003	
20243	079	

Indirizzo RAM	Codice decimale	Mnemonico
20244	237	SBC HL, BC
20245	066	
20246	205	CALL 20297
20247	073	
20248	079	
20249	237	LD BC, (20227)
20250	075	
20251	003	
20252	079	
20253	009	ADD HL, BC
20254	009	ADD HL, BC
20255	205	CALL 20297
20256	073	
20257	079	
20258	033	LD HL, 692
20259	180	
20260	002	
20261	237	LD BC, (20225)
20262	075	
20263	001	
20264	079	
20265	237	SBC HL, BC
20266	066	
20267	034	LD (20225), HL
20268	001	
20269	079	
20270	042	LD HL, (16396)
20271	012	
20272	064	
20273	237	LD BC, (20225)
20274	075	
20275	001	
20276	079	
20277	009	ADD HL, BC
20278	237	LD BC, (20227)
20279	075	
20280	003	
20281	079	
20282	237	SBC HL, BC
20283	066	
20284	205	CALL 20297
20285	073	
20286	079	
20287	237	LD BC (20227)
20288	075	
20289	003	
20290	079	
20291	009	ADD HL, BC
20292	009	ADD HL, BC

Indirizzo RAM	Codice decimale	Mnemonico
20293	205	CALL 20297
20294	073	
20295	079	
20296	201	RETURN
20297	114	LD (HL), D
20298	035	INC HL
20299	114	LD (HL), D
20300	001	LD BC, 32
20301	032	
20302	000	
20303	009	ADD HL, BC
20304	114	LD (HL), D
20305	035	INC HL
20306	114	LD (HL), D
20307	001	LD BC, 34
20308	034	
20309	000	
20310	237	SBC HL, BC
20311	066	
20312	201	RETURN

(subroutine: 20297–20312)

```
  10 REM  CALEIDOSCOPIO
  20 REM  ----------------
 100 GOSUB 270
 110 SLOW
 120 RAND
 130 LET T=49+66*INT (5*RND)
 140 LET P=1+2*INT (5*RND)
 150 POKE 20225,T-(INT (T/256))*
256
 160 POKE 20226,INT (T/256)
 170 POKE 20227,P
 180 LET X=INT (192*RND)
 190 IF X>63 THEN LET X=X+64
 200 IF X>191 THEN LET X=0
 210 POKE 20231,X
 220 LET M=USR 20230
 230 FOR R=1 TO 1
 240 NEXT R
 250 IF INKEY$="S" THEN GOTO 370
 260 GOTO 130
 270 FAST
 290 POKE 20228,0
 300 LET A$="022 000 042 012 064
237 075 001 079 009 237 075 003
```

Figura 123 (Continua)

```
079 237 066 205 073 079 237 275
003 079 009 009 205 073 079 033
180 002 237 075 001 079 237 066
034 001 079 042 012 064 237 075
001 079 009 237 075 003 079 237
066 205 073 079 237 075 003 079
009 009 205 073 079 201 114 035
114 001 032 000 009 114 035 114
001 034 000 237 066 201"
310 LET A=20230
320 FOR L=1 TO LEN A$ STEP 4
330 POKE A,VAL A$(L TO L+2)
340 LET A=A+1
350 NEXT L
360 RETURN
```

Figura 123 (Fine)

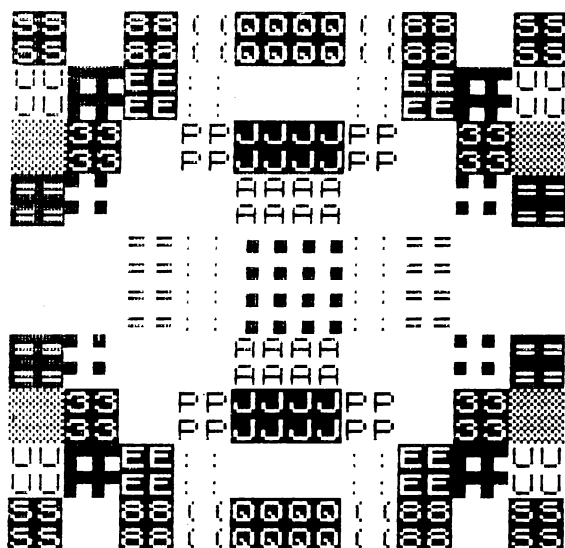



Figura 124

```
180 LET X=INT (31*RND)
190 IF X>10 THEN LET X=X+117
200 IF X>138 THEN LET X=0
```

Figura 125

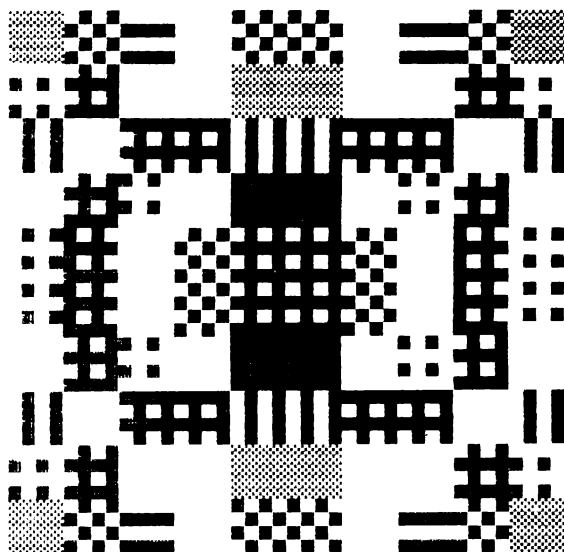



Figura 126

RoMbOsPiRaLe SLOW

Anche senza ricorrere a delle apposite routines in linguaggio macchina, si possono ottenere delle buone animazioni. Una di queste è ottenuta dal programma Rombo-spirale (fig. 127).

```
  10 REM ROMBOSPIRALE
 100 SLOW
 110 LET C=1
 120 GOSUB 370
 130 FOR P=1 TO 11
 140 PRINT AT 12-P,P+5;CHR$ X
 150 NEXT P
 160 FOR N=1 TO 11 STEP 2
 170 GOSUB 370
 180 FOR P=1 TO 12-N
 190 PRINT AT -1+P+N,15+P;CHR$ X
 200 NEXT P
 210 GOSUB 370
 220 FOR P=1 TO 12-N
 230 PRINT AT 10+P,28-P-N;CHR$ X
```

Figura 127 (Continua)

```
 240  NEXT P
 250  GOSUB 370
 260  FOR P=1 TO 11-N
 270  PRINT AT 23-P-N,17-P;CHR$ X
 280  NEXT P
 290  GOSUB 370
 300  FOR P=1 TO 11-N
 310  PRINT AT 13-P,5+P+N;CHR$ X
 320  NEXT P
 330  NEXT N
 340  LET C=C*(-1)
 350  LET X=0
 360  GOTO 120
 370  IF C=1 THEN LET X=1+INT (12
7*RND)
 380  IF X>63 THEN LET X=X+64
 390  RETURN
```

Figura 127 (Fine)

Tale programma disegna e cancella alternativamente in continuazione, una spirale a forma di rombo formata da caratteri scelti casualmente.
La spirale viene disegnata o cancellata partendo dall'esterno verso l'interno. La spirale appena completata appare, per esempio, come in fig. 128.

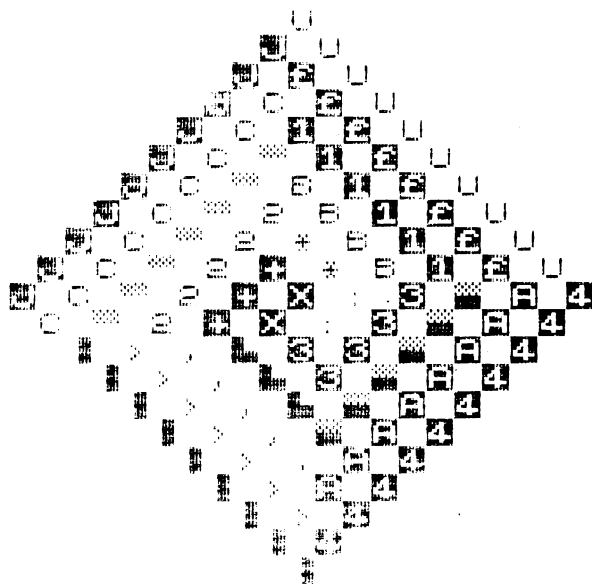



Figura 128

Un'altra animazione unicamente in BASIC è l'esplosione di un oggetto colpito da un proiettile creata dal programma di fig. 129.

```
 10 REM  ANIMAZIONE ESPLOSIONE
100 SLOW
110 PRINT AT 10,15;"█"
120 PRINT AT 11,15;"█"
130 PAUSE 50
135 POKE 9999,139
140 FOR A=1 TO 30
150 PLOT A,21
160 UNPLOT A-1,21
170 NEXT A
180 UNPLOT 30,21
190 UNPLOT 32,21
195 POKE 9999,7
200 PAUSE 10
210 FOR K=0 TO 5
220 UNPLOT 31-K,20-K
230 UNPLOT 31-K,21
240 UNPLOT 31-K,22+K
250 UNPLOT 32,20-K
260 UNPLOT 32,22+K
270 UNPLOT 33+K,20-K
280 UNPLOT 33+K,21
290 UNPLOT 33+K,22+K
295 POKE 9999,15
300 IF K=5 THEN GOTO 400
310 PLOT 30-K,19-K
320 PLOT 30-K,21
330 PLOT 30-K,23+K
340 PLOT 32,19-K
350 PLOT 32,23+K
360 PLOT 34+K,19-K
370 PLOT 34+K,21
380 PLOT 34+K,23+K
385 POKE 9999,11
390 NEXT K
400 STOP
```

Figura 129

Le figure da 130 a 136 mostrano in sequenza alcune fasi dell'esplosione. Il programma ha anche quattro istruzioni (135, 195, 295, 385) per avere anche dei suoni una

volta che sarà collegata la scheda musicale che è presentata in dettaglio nel capitolo sulla musica.

Figura 130

Figura 131

Figura 132

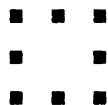



Figura 133

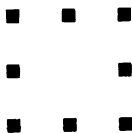



Figura 134

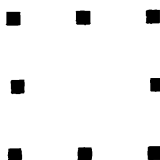



Figura 135

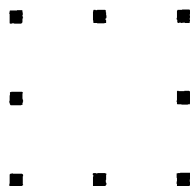



Figura 136

Programmazione universale animazione con PRINT AT — SLOW

Una complessa animazione può richiedere decine di linee di programma con PRINT AT.
Come già visto nel capitolo sulla Grafica, una lunga serie di linee con PRINT AT può essere condensata in un programma più breve.
Il programma universale di animazione di fig. 137 è in gran parte uguale al programma universale per PRINT AT di fig. 26 con l'aggiunta, però di alcune linee per aggiungere la funzione del Tempo.
Anche nel programma di fig. 137 i codici per le coordinate delle linee (in L$) e delle colonne (in C$) sono quelli elencati nella tabella 1. I caratteri sono sempre nella stringa X$ mentre i Tempi tra l'esecuzione di un PRINT AT e il successivo sono contenuti nella stringa T$ e codificati secondo la tabella 2; tali tempi sono in una scala arbitraria, vale a dire £ non corrisponde ad un secondo ma solo al tempo più breve, e allo stesso modo X rappresenta solo il tempo più lungo.
Se si ha bisogno di un disegno complesso insieme ad una lunga animazione si può

```
  10 REM  PROGRAMMA UNIVERSALE
  20 REM  ANIMAZIONE CON
  30 REM  PRINT AT E IN SLOW
  66 FOR H=5 TO 14
  77 PRINT AT H,10;"█    █"
  88 NEXT H
  99 PRINT AT 15,10;"██████"
 100 SLOW
 110 REM  CODICI LINEE IN L$
 120 LET L$="0011223344556677889
9AABBCCDDEEDDCCBBAA9988776655443
3221100"
 130 REM  CODICI COLONNE IN C$
 140 LET C$="778899AABBCCDDEEDDC
CBBCCDDEEDDCCBBCCDDEEDDCCBBCCDDE
EFFGGHH"
 150 REM  CARATTERI IN X$
 160 LET X$="H O U 8 H ▒ ▒ ▀ ▒ █
▘▖ ▝▗ ▒ ▄ ▀ █ $ £ * ▒ ▄ ▀ + X H O
? . / "
 170 REM  TEMPI IN T$
 180 LET T$=">>>>>>>>>>£££££££££
£££££££££££££££££££££££££££££££££
£$$$$$"
 190 IF LEN L$<>LEN C$ OR LEN C$
<>LEN X$ OR LEN X$<>LEN T$ THEN
GOTO 260
 200 FOR P=1 TO LEN X$
 210 PRINT AT CODE L$(P TO )-28,
CODE C$(P TO )-28;X$(P TO P)
 220 FOR T=1 TO CODE T$(P TO )-1
1
 230 NEXT T
 240 NEXT P
 250 STOP
 260 PRINT AT 0,10;"ERRORE"
 270 PAUSE 100
 280 FAST
 290 LIST
```

Figura 137

inserire a questo programma di animazione in SLOW dopo il programma per disegnare di fig. 26, in FAST.
Al programma universale di fig. 137 sono state aggiunte alcune linee (66, 77, 88, 99) e nelle stringhe L$, C$, X$, T$ sono stati inseriti i codici e i caratteri per un esempio di animazione, esempio che mostra un oggetto in movimento che entra angolato in un contenitore, rimbalza sulle pareti cambiando continuamente forma ed esce nel

Tabella 1

CODICI LINEE E COLONNE

L/C		COD.	L/C		COD.	C		COD.
0	>	0	11	>	B	22	>	M
1	>	1	12	>	C	23	>	N
2	>	2	13	>	D	24	>	O
3	>	3	14	>	E	25	>	P
4	>	4	15	>	F	26	>	Q
5	>	5	16	>	G	27	>	R
6	>	6	17	>	H	28	>	S
7	>	7	18	>	I	29	>	T
8	>	8	19	>	J	30	>	U
9	>	9	20	>	K	31	>	V
10	>	A	21	>	L			

Tabella 2

CODICI TEMPI

T		COD.	T		COD.	T		COD.
1	>	£	19	>	2	37	>	K
2	>	$	20	>	3	38	>	L
3	>	:	21	>	4	39	>	M
4	>	?	22	>	5	40	>	N
5	>	(	23	>	6	41	>	O
6	>	)	24	>	7	42	>	P
7	>	>	25	>	8	43	>	Q
8	>	<	26	>	9	44	>	R
9	>	=	27	>	A	45	>	S
10	>	+	28	>	B	46	>	T
11	>	-	29	>	C	47	>	U
12	>	*	30	>	D	48	>	V
13	>	/	31	>	E	49	>	W
14	>	;	32	>	F	50	>	X
15	>	,	33	>	G			
16	>	.	34	>	H			
17	>	0	35	>	I			
18	>	1	36	>	J			

Figura 138

senso opposto. La linea 160 può essere modificata come indicato in fig. 138 (con 29 spazi inversi alternati a 29 spazi normali) per fare in modo che l'oggetto in movimento abbia sempre la stessa forma.

Programma universale animazione con PLOT-UNPLOT — SLOW

Se si vuole una risoluzione quattro volte maggiore e se non è necessario inserire dei caratteri, si può realizzare un'animazione con PLOT e UNPLOT utilizzando il programma di fig. 139.
Tutte le istruzioni di animazione sono inserite in un'unica stringa (C$) ed ogni istruzione è costituita da sei caratteri che formano due numeri da due caratteri separati da un segno + oppure − e seguiti da uno spazio che separa un'istruzione dall'altra. Il primo ed il secondo numero di ogni istruzione rappresentano rispettivamente le coordinate X e Y del pixel da plottare (se tra i numeri c'e un +) o da unplottare (se tra i numeri c'è un −).

```
  10 REM  PROGRAMMA UNIVERSALE
  20 REM  ANIMAZIONE CON PLOT
  30 REM  UNPLOT E IN SLOW
 100 SLOW
 110 REM  COORDINATE IN C$ :
 120 REM  1■ VALORE - COORD. <X>
 130 REM  2■ VALORE - COORD. <Y>
 140 REM  AL CENTRO SCELTA MODO
 150 REM      + PLOT    - UNPLOT
 160 LET C$="09+30 07+32 08+32 1
0+32 12+32 14+32 16+32 18+32 19+
32 09-30 09+31 09-31 09+32 09+33
 09-33 10+33 10-33 11+33 11-33 1
1+32 11+31 11-31 12+31 12-31 13+
31 13-31 13+32 13+33 13-33 14+33
 14-33 15+33 15-33 15+32 15+31 1
5-31 16+31 16-31 17+31 17-31 17+
32 17+33 17-33 17+34"
 170 FOR P=1 TO LEN C$ STEP 6
 180 IF CODE C$(P+2 TO )=21 THEN
 PLOT VAL C$(P TO P+1),VAL C$(P+
3 TO P+4)
 190 IF CODE C$(P+2 TO )=22 THEN
 UNPLOT VAL C$(P TO P+1),VAL C$(
P+3 TO P+4)
 200 NEXT P
```

Figura 139

Anche in questo caso la stringa con le istruzioni contiene un esempio di animazione che non ha un'utilità pratica ma che mostra il funzionamento del programma.

Figura 140

Figura 141

Figura 142

L'animazione (di cui nelle figure 140, 141, 142 si può vedere l'inizio, una fase intermedia e la fine) rappresenta una sorta di muro con dei fori che vengono attraversati e riempiti da un pixel in movimento.

PROGRAMMI PER INTERFACCIA

Circuito di interfaccia per ZX80-ZX81

Con solo due integrati (due comuni TTL-LS) e un costo totale inferiore a 5000 lire si può realizzare un circuito di Interfaccia ad 8 uscite che permette di collegare uno ZX81 o un ZX80 al mondo esterno consentendo di realizzare numerosi programmi di automazione, di controllo, di robotica, ecc. altrimenti non possibili con il solo computer.
Oltre a ciò il circuito di Interfaccia è indispensabile per pilotare il circuito musicale esposto nel prossimo capitolo, così come altri circuiti realizzati da ciascun possessore di un ZX.

IMPORTANTE - L'interfaccia va collegata tramite dei fili direttamente al circuito stampato dello ZX.
TENERE PRESENTE CHE UNA ERRATA COSTRUZIONE DEL CIRCUITO O UN ERRATO COLLEGAMENTO DEI FILI AL CIRCUITO STAMPATO PUO' CAUSARE DANNI AL COMPUTER, per cui se vi occupate di programmazione ma non di circuiti elettronici, è preferibile che facciate realizzare o controllare il tutto ad un Tecnico che abbia già esperienza di montaggi e saldature.

Lo schema del circuito di Interfaccia è in fig. 143 ed usa, come già detto, due integrati: un 74LS373 che contiene 8 memorie LATCH ed un 74LS27 che contiene tre NOR a tre ingressi utilizzati per la decodifica dell'indirizzo di memoria.
Le configurazioni dei due integrati sono in fig. 144 e 145.
I quindici ingressi del circuito di Interfaccia vanno collegati, tramite quindici spezzoni di filo sottile ad un capo, a quindici dei quaranta piedini dello Z80. La saldatura va effettuata sul circuito stampato dal lato delle saldature. In figura 146 è mostrato lo Z80 **visto da sotto** con indicati i quindici piedini ai quali vanno saldati i fili provenienti dall'Interfaccia. Per i collegamenti tra computer e Interfaccia il filo più adatto può essere quello per WIRE-WRAP che è appunto molto sottile e ad un capo.
Prima di richiudere il computer controllare di avere collegato i fili esattamente e che i fili stessi siano saldati bene e non si stacchino.
Se avete usato dei fili abbastanza sottili potete farli uscire all'esterno tramite uno dei fori già esistenti sui due ZX.

NUMERI PINS Z80
INGRESSI
USCITE
+5V
D0
D1
D2
D3
D4
D5
D6
D7
A15
A14
A13
MREQ
WR
GND
74LS373
74LS27
LED gialli
220
1N4148
1,5k
A
B
C
D
E
F
G
H
+5V
GND
AL CONNETTORE A MORSETTI
OFF
ON
SW1
interruttore on/off LED
1/4 W - 5%

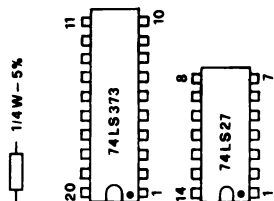



Figura 143

OCTAL D-TYPE LATCHES

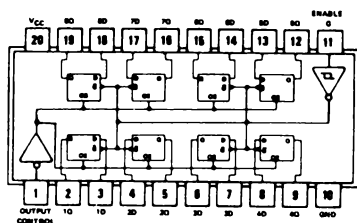
3 STATE OUTPUTS
COMMON OUTPUT CONTROL
COMMON ENABLE

74LS373

Figura 144

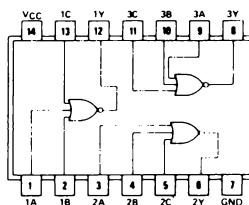
TRIPLE 3-INPUT
POSITIVE-NOR GATES

74LS27

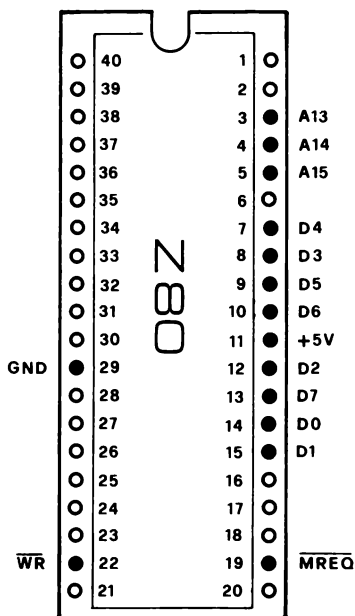
Figura 145

C.I. VISTO DA SOTTO IL
CIRCUITO STAMPATO
(LATO SALDATURE)

Figura 146

Le dieci uscite dell'Interfaccia (A......H, +5V, GND) vanno collegate ad un connettore del tipo visibile in fig. 147 al quale potranno essere collegati i circuiti esterni come ad esempio la scheda musicale.
Il consumo del circuito senza carico e con i LED non inseriti è di 22 mA, con i LED inseriti e accesi può arrivare fino a 50 mA.

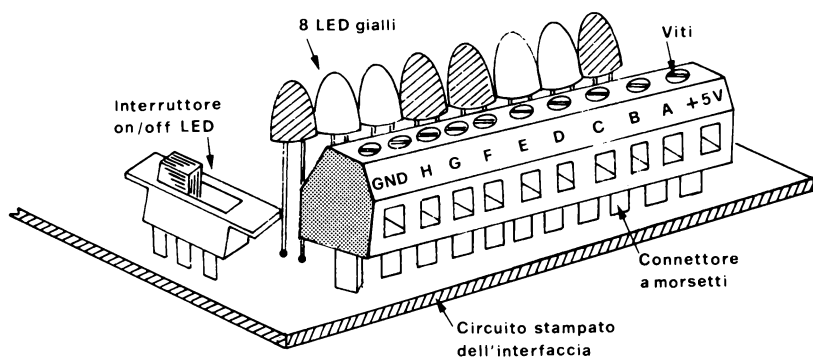



Figura 147

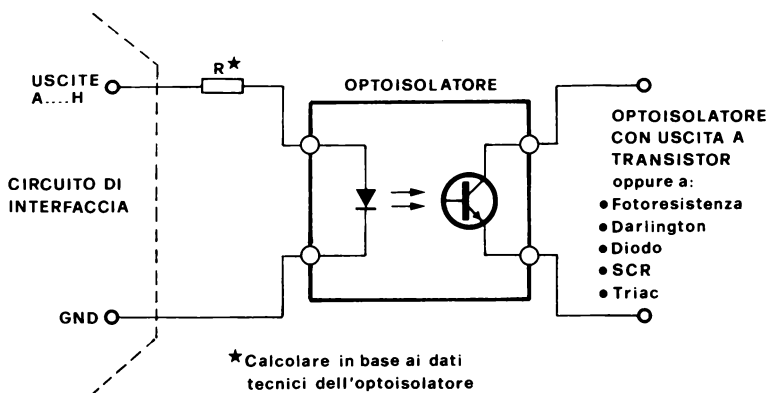



Figura 148

Il funzionamento della scheda può essere verificato, con l'interruttore per l'inserimento dei LED in posizione ON, eseguendo il seguente programma:

```
10 FAST (solo su ZX-81)
20 POKE 8888,0
30 PAUSE 25
40 POKE 8888,255
50 PAUSE 25
60 GOTO 20
```

Dopo avere premuto RUN e NEWLINE gli otto LED devono accendersi e spegnersi contemporaneamente al ritmo di una volta al secondo.
Se qualcuno dei LED non si accende o rimane sempre acceso ricontrollare innanzitutto le linee 20 e 40 del programma, se tali linee sono corrette allora la causa può essere dovuta ad un LED difettoso, ad una interruzione, ad un corto circuito,

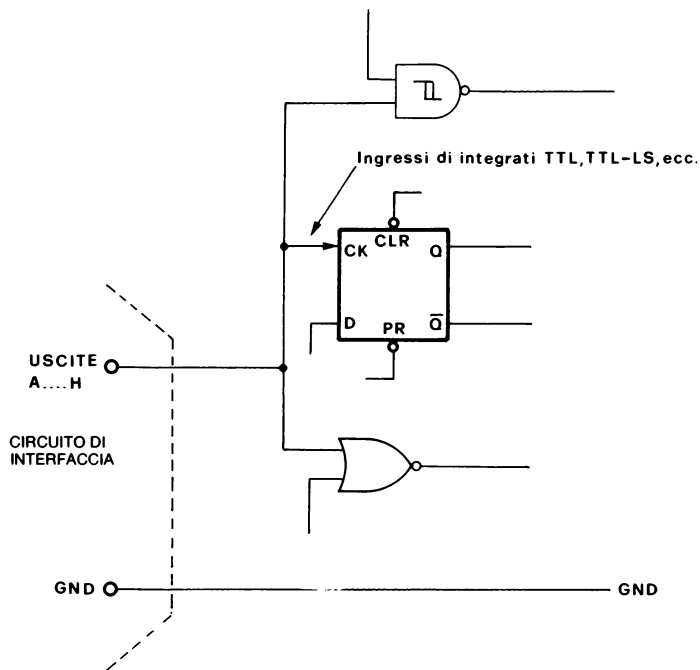



Figura 149

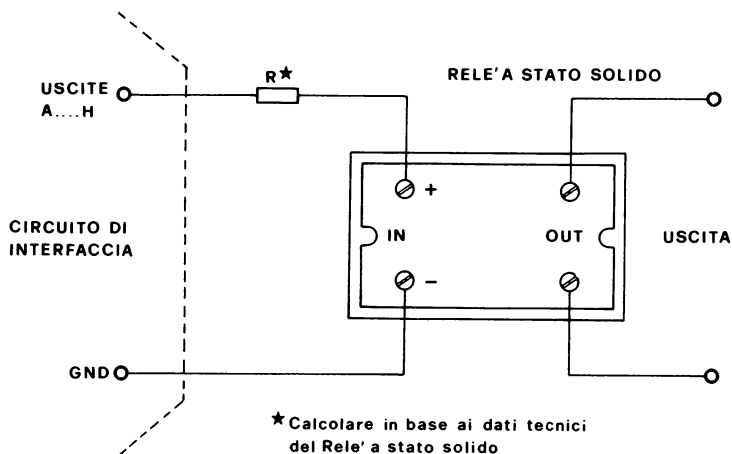



Figura 150

all'errato collegamento tra computer e Interfaccia o anche a un integrato difettoso.
A seconda del programma utilizzato e dei circuiti collegati alle otto uscite, il circuito di Interfaccia può essere usato in moltissime applicazioni.
Alle otto uscite TTL A....H possono essere collegati vari tipi di circuiti. Si possono collegare, per esempio, degli optoisolatori (fig. 148) con ingresso a LED e uscita a fotoresistenza oppure a fototransistor, a fotodarlington, a fotodiodo, a fotoSCR o a fotoTRIAC. L'uso di optoisolatori consente di isolare il computer e l'Interfaccia dall'apparecchio che devono pilotare.
Naturalmente alle otto uscite dell'Interfaccia si possono collegare gli ingressi di circuiti logici TTL, TTL-LS, TTL-S, ecc. (fig. 149) il numero di ingressi collegabili a ciascuna uscita dipende dal tipo di logica usata. Se con l'Interfaccia si vogliono pilotare carichi ad alta tensione e/o ad alta corrente, si possono collegare ad essa

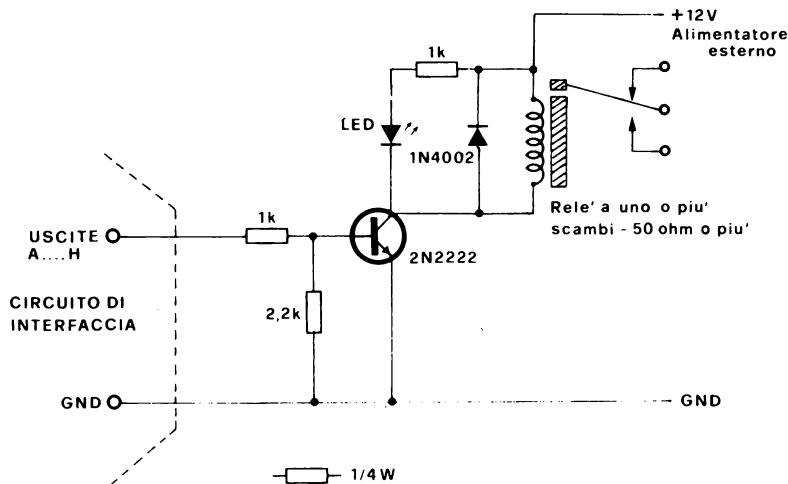



Figura 151

dei Relè a Stato Solido (fig. 150) oppure dei normali Relè Elettromagnetici usando per esempio il circuito di fig. 151. Poichè i Relè assorbono molta corrente occorre alimentarli con un alimentatore esterno e non tramite il computer. Sempre nel circuito di fig. 151 si possono utilizzare Relè funzionanti a tensioni diverse da 12V, oppure usare carichi diversi da un Relè, così come anche il transistor può essere sostituito con altri tipi oppure con un darlington o un VMOS.
Infine, l'interfaccia è indispensabile per pilotare la scheda musicale esposta nel prossimo capitolo.
Nel collegare l'Interfaccia ad un qualsiasi carico occorre tenere presente che il circuito è alimentato dal computer e che, perciò, occorre limitare la corrente assorbita dal suo alimentatore.
Dalle uscite A....H e +5V insieme, si può assorbire un totale di 100-150 mA al massimo (di cui 10-15 mA massimo da ciascuna delle otto uscite TTL e il resto

dall'uscita +5V); tale corrente totale può arrivare anche a 200-250 mA se si usa l'alimentatore Sinclair più grande, alimentatore che è comunque indispensabile se, oltre all'Interfaccia, vengono collegati al computer la stampante e/o l'espansione RAM. Tenete presente che un eccessivo assorbimento di corrente surriscalda l'integrato regolatore a 5V e di conseguenza anche l'interno del computer.
L'interfaccia viene comandata dal computer tramite una semplice istruzione in BASIC (POKE). La POKE può essere compresa tra POKE 8192,n e POKE 16383,n si suggerisce, comunque, di usare o la POKE 8888,n oppure la POKE 9999,n in

Tabella 3

POKE 8888,	USCITA ATTIVA ("1" logico TTL)	
1	A	(DØ)
2	B	(D1)
4	C	(D2)
8	D	(D3)
16	E	(D4)
32	F	(D5)
64	G	(D6)
128	H	(D7)

POKE: *minimo* 8888,Ø *massimo* 8888,255
con POKE 8888,Ø Tutte le uscite DISATTIVATE ("0" logico TTL)

esempio: POKE 8888,75
USCITE ATTIVE: A, B, D, G
(1+2+8+64 = 75)

quanto sono più facili da ricordare e usano solo quattro numeri (in questo e nei prossimi capitoli viene usata la POKE 8888 nei programmi che usano solo l'Interfaccia, e la POKE 9999 nei programmi che usano la scheda musicale, questo per indicare quando la POKE viene usata per produrre un suono). Nella POKE il numero "n" va compreso tra 0 e 255, con "n" uguale a 0 tutte e otto le uscite sono allo "0" logico, con "n" uguale a 255 tutte e otto le uscite sono all'"1" logico.
Le combinazioni richieste possono essere ottenute tramite la Tabella 3; per esempio, se si vogliono attivare le uscite B, E, F, H, occorre sommare i corrispondenti numeri della Tabella 3 (2+16+32+128 = 178) ed effettuare l'istruzione POKE 8888,178. Allo stesso modo si può ottenere qualsiasi combinazione di uscite attive e non attive.

Nei programmi in cui occorra una maggiore velocità l'Interfaccia può essere comandata anche da un istruzione in Linguaggio Macchina, anche in questo caso scrivendo un numero (n) in uno qualsiasi degli indirizzi compresi tra 8192 e 16383.

Programmi

L'uso dell'Interfaccia è legato soprattutto alle applicazioni ideate da ciascun utilizzatore. In sostanza ognuno potrà collegare il computer tramite l'Interfaccia a delle attrezzature, dei servomeccanismi, degli strumenti, ecc. che utilizza nel proprio lavoro o nel proprio hobby, questo allo scopo di rendere automatici o più veloci alcuni processi o di avere nuove funzioni e applicazioni rese possibili dalla potenza e dalla velocità elaborativa del computer.
Le applicazioni esposte qui di seguito comprendono un programmatore universale utilizzabile in molti casi pratici, alcuni programmi di esempio per produrre degli effetti di luce ed infine un tasto elettronico per telegrafia.

Programma universale per circuito di Interfaccia — FAST

Il programmatore universale (fig. 152) può eseguire una serie di istruzioni, alternate a delle pause di durata prestabilita.
La stringa D$ (linea 110) contiene in forma decimale le istruzioni sperate da uno

```
  10 REM PROGRAMMATORE
  20 REM UNIVERSALE PER
  30 REM CIRCUITO DI INTERFACCIA
 100 FAST
 102 REM  D$ CONTIENE I CODICI
 104 REM  PER PILOTARE I
 106 REM  DISPOSITIVI COLLEGATI
 108 REM  ALL■ INTERFACCIA
 110 LET D$="064 155 200 015 014
 100 128 011 255 064 200 088"
 112 REM  P$ CONTIENE LE PAUSE
 114 REM  DA EFFETTUARE TRA UNA
 116 REM  POKE E LA SUCCESSIVA
 120 LET P$="100 100 150 500 050
 010 800 160 080 200 100 225"
 130 FOR A=1 TO LEN D$ STEP 4
 140 POKE 8888,VAL D$(A TO A+2)
 160 PAUSE VAL P$(A TO A+2)
 180 NEXT A
 190 GOTO 130
```

Figura 152

ESEMPIO PROGRAMMA 12 PASSI

P$	D$	A	B	C	D	E	F	G	H
		DISPOSITIVI							
	064							■	
100 PAUSA 2 S.									
	155	■	■		■	■			■
100 PAUSA 2 S.									
	200				■			■	■
150 PAUSA 3 S.									
	015	■	■	■	■				
500 PAUSA 10 S.									
	014		■	■	■				
050 PAUSA 1 S.									
	100			■			■	■	
010 PAUSA 0.2 S.									
	128								■
800 PAUSA 16 S.									
	011	■	■		■				
160 PAUSA 3.2 S.									
	255	■	■	■	■	■	■	■	■
080 PAUSA 1.6 S.									
	064							■	
200 PAUSA 4 S.									
	200				■			■	■
100 PAUSA 2 S.									
	088				■	■		■	
225 PAUSA 4.5 S.									

Figura 153

spazio; il numero decimale di ogni istruzione (che deve essere sempre di tre cifre) può essere ricavato utilizzando la Tabella 3 in base ai dispositivi (collegati alle uscite A....H) che l'istruzione stessa deve attivare o non attivare in quel dato momento.

```
145 LET T=VAL P$(A TO A+2)
150 FOR M=1 TO 50
160 PAUSE T
170 NEXT M
```

Figura 154

```
145 LET T=VAL P$(A TO A+2)
150 FOR M=1 TO 3000
160 PAUSE T
170 NEXT M
```

Figura 155

```
 10 REM  LUCI ROTANTI
100 FAST
110 LET T=1
120 POKE 8888,1
130 PAUSE T
140 POKE 8888,2
150 PAUSE T
160 POKE 8888,4
170 PAUSE T
180 IF INKEY$="6" THEN LET T=T+
1
190 IF T>1 AND INKEY$="7" THEN
LET T=T-1
200 GOTO 120
```

Figura 156

La stringa P$ (linea 120) contiene, invece, le pause che il computer deve inserire tra l'esecuzione di una istruzione e la successiva.
Il programma di fig. 152 contiene nelle stringhe D$ e P$ un esempio di programmazione lungo 12 passi, esempio esposto più chiaramente in fig. 153.
La prima istruzione di D$ è Ø64 che attiva il dispositivo collegato all'uscita G dell'Interfaccia; la prima istruzione diP$ è 1ØØ che introduce una pausa di 2 secondi, dopodichè viene eseguita la seconda istruzione di D$ (155) che attiva altri dispositivi, seguita da un'altra Pausa e così via.
La sequenza appena esposta può essere, ad esempio, quella di un processo di automazione.

+12V
1N4002
2N2222
1k
2,2k
A
B
C
GND
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
220V~
DAL CIRCUITO DI INTERFACCIA
1/4 W – 5%
L1----L12 a 220V~ (anche piu'di dodici)
Rele' 12V—100ohm min.

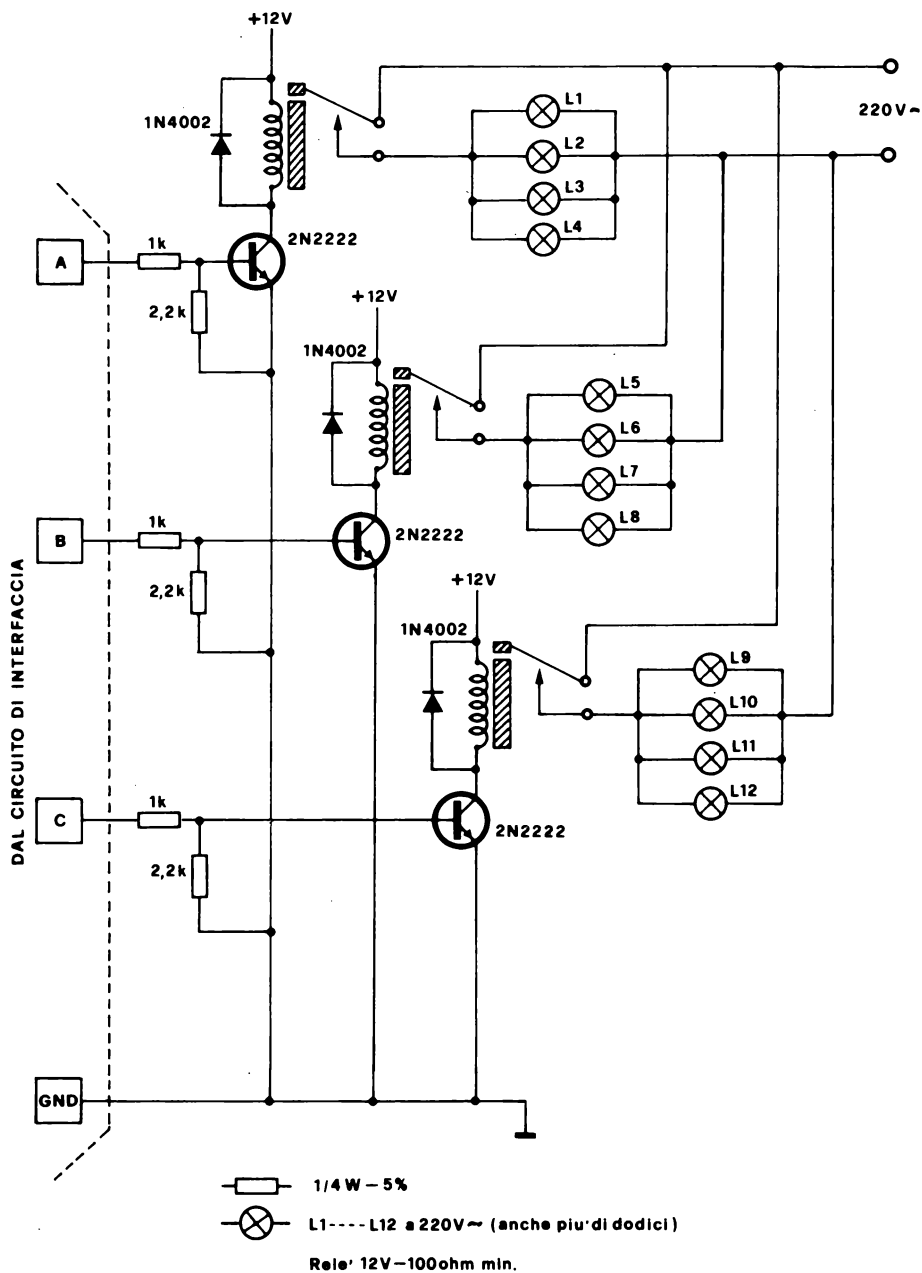



Figura 157

Senza collegare dei dispositivi alle uscite dell'Interfaccia, il funzionamento del programma può essere messo in evidenza dall'accendersi e spegnersi degli otto LED.
Naturalmente la sequenza può esser costituita da un numero maggiore o minore delle dodici istruzioni dell'esempio purchè il numero di istruzioni contenute in D$ e in P$ sia uguale, vale a dire che le due stringhe abbiano la stessa lunghezza.
I numeri delle istruzioni di pausa contenute nella stringa P$ possono essere compresi tra 000 e 999, con una pausa massima, quindi, di circa 20 secondi. Inserendo nel programma le linee di fig. 154 i numeri contenuti in P$ non saranno più cinquantesimi di secondo ma secondi (da 000 a 999). Infine, inserendo nel programma le linee di fig. 155 i numeri contenuti in P$ corrisponderanno a minuti.
Tenete presente che i computer Sinclair non usano come clock un quarzo ma un filtro ceramico che ha una precisione inferiore, per cui anche i minuti o i secondi o i cinquantesimi di secondo delle istruzioni di pausa contenute in P$ avranno una precisione inferiore a sistemi quarzati.

Luci rotanti FAST

Il programma di luci rotanti (fig. 156) è il primo di quattro effetti di luce che utilizzano il circuito di Interfaccia. Per usare il programma occorre collegare a tre delle uscite dell'Interfaccia (A, B, C) tre relè (fig. 157), con alimentatore separato dal computer, ciascuno dei quali pilota l'accensione di due o più lampadine anch'esse alimentate esternamente e disposte come in fig. 158.

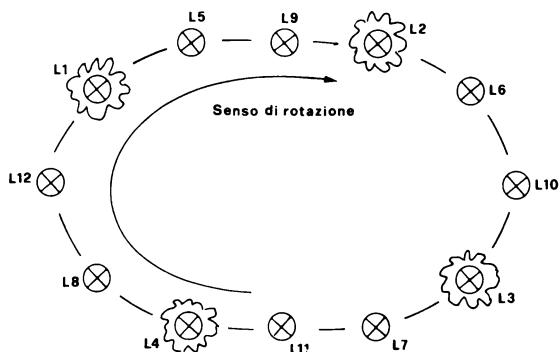



Figura 158

La velocità di rotazione che parte con il massimo può essere diminuita tenendo premuto il tasto "6" e di nuovo aumentata tenendo premuto il tasto "7".
Il senso di rotazione può essere invertito senza toccare la sistemazione delle lampadine semplicemente spostando il contenuto della linea 160 nella linea 120 ed il contenuto della linea 120 nella linea 160.

Il programma di fig. 159 comanda l'accensione e lo spegnimento casuale di otto lampadine, collegate alle otto uscite dell'Interfaccia tamite otto circuiti come quello di fig. 160.

```
  10 REM  LUCI CASUALI
 100 FAST
 110 LET T=1
 120 RAND
 130 LET N=INT (256*RND)
 140 POKE 8888,N
 150 IF INKEY$="6" THEN LET T=T+
1
 160 IF T>1 AND INKEY$="7" THEN
LET T=T-1
 170 PAUSE T
 180 GOTO 130
```

Figura 159

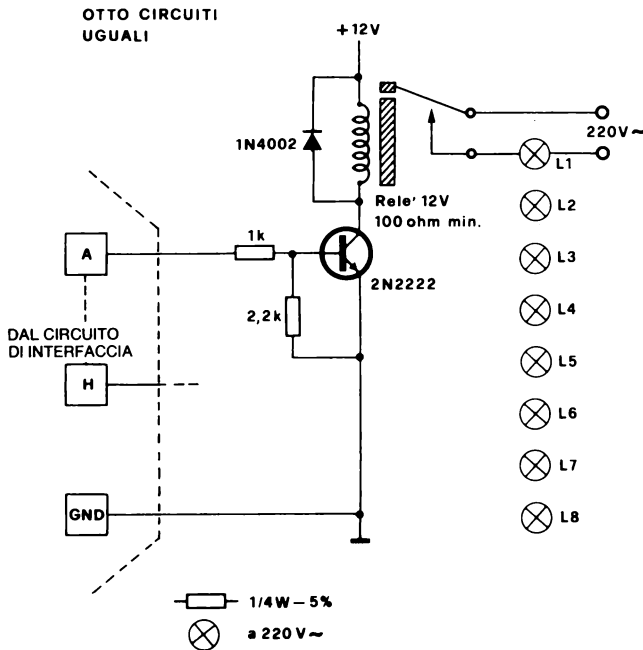



Figura 160

Come nel programma precedente, anche in questo caso, la velocità di cambiamento si può diminuire premendo il tasto "6" e di nuovo aumentare premendo il tasto "7", tale velocità si può, inoltre, rendere casuale modificando il programma come in fig. 161.

```
  10 REM   LUCI CASUALI
  20 REM   CON RITARDO CASUALE
 100 FAST
 110 LET T=1
 120 RAND
 130 LET N=INT (256*RND)
 140 POKE 8888,N
 160 PAUSE INT (10*RND)
 180 GOTO 130
```

Figura 161

Senza collegare alle uscite i relè e le lampadine il funzionamento di questo e dei prossimi due programmi può essere messo in evidenza anche dai LED dell'Interfaccia.

Luci a riempimento e svuotamento FAST

Sempre con 8 lampadine alle otto uscite dell'Interfaccia si può ottenere un effetto di riempimento e svuotamento graduale della linea di luci facendo girare il programma di fig. 162.

```
  10 REM  LUCI A RIEMPIMENTO
  20 REM  E SVUOTAMENTO
 100 FAST
 110 LET T=1
 120 LET M=1
 130 IF M=1 THEN FOR A=0 TO 8
 140 IF M=-1 THEN FOR A=8 TO 0 S
TEP -1
 150 POKE 8888,2**A-1
 160 IF INKEY$="6" THEN LET T=T+
1
 170 IF T>1 AND INKEY$="7" THEN
LET T=T-1
 180 FOR D=1 TO T
 190 NEXT D
 200 NEXT A
 210 LET M=M*-1
 220 GOTO 130
```

Figura 162

Anche quì la velocità può essere modificata premendo i tasti 6 e 7. Si può anche ottenere un effetto di solo svuotamento modificando le linee 130 e 210 nel modo indicato in fig. 163, oppure un effetto di solo riempimento modificando le stesse linee come indicato in fig. 164.

```
130 FOR A=8 TO 0 STEP -1

210 POKE 8888,255
```

Figura 163

```
130 FOR A=0 TO 8

210 POKE 8888,0
```

Figura 164

Punto luminoso mobile FAST

L'ultimo programma per effetti di luce (fig. 165) fa muovere un punto luminoso lungo la linea di otto lampadine collegate alle otto uscite dell'Interfaccia.
Di nuovo i tasti 6 e 7 regolano la velocità di movimento.

```
  10 REM  PUNTO LUMINOSO MOBILE
 100 FAST
 110 LET T=1
 120 FOR A=0 TO 7
 130 POKE 8888,2**A
 140 IF INKEY$="6" THEN LET T=T+
1
 150 IF T>1 AND INKEY$="7" THEN
LET T=T-1
 160 FOR D=1 TO T
 170 NEXT D
 180 NEXT A
 190 POKE 8888,0
 200 GOTO 120
```

Figura 165

Tasto automatico per telegrafia FAST

Sempre utilizzando l'Interfaccia si può realizzare un Tasto automatico per telegrafia da 3-30 parole/minuto facendo girare il programmà di fig. 166.

```
  10 REM  TASTO AUTOMATICO
  20 REM    PER TELEGRAFIA
  30 REM 3-30 PAROLE/MINUTO
  40 REM ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
  50 REM   <C> = ........
  60 REM   <B> = --------
  70 REM ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
 100 FAST
 110 PRINT "INSERIRE VELOCITA■ "
 120 PRINT "1--9 PUNTI AL SECOND
O"
 130 PRINT "E PREMERE NEWLINE"
 140 INPUT P
 150 LET P=INT P
 160 IF P<1 OR P>9 THEN GOTO 140
 170 LET T=INT (25/P)
 180 LET TT=3*T
 190 CLS
 200 IF INKEY$="C" THEN GOTO 230
 210 IF INKEY$="B" THEN GOTO 280
 220 GOTO 200
 230 POKE 9999,235
 240 PAUSE T
 250 POKE 9999,15
 260 PAUSE T
 270 GOTO 200
 280 POKE 9999,235
 290 PAUSE TT
 300 POKE 9999,15
 310 PAUSE T
 320 GOTO 200
```

Figura 166

All'inizio occorre scegliere una delle nove velocità (fig. 167) inserendo il numero di punti al secondo (da 1 a 9) quindi, premuto NEWLINE, il computer produrrà una serie di punti ogni volta che si preme il Tasto "C" e una serie di linee ogni volta che si prme il Tasto "B"; i punti e le linee prodotti sono autocompletanti e la durata è in proporzione tra loro e in rapporto alla velocità scelta.

```
INSERIRE VELOCITA■
1--9 PUNTI AL SECONDO
E PREMERE NEWLINE
```

Figura 167

Come si è detto si possono inserire nove velocità, togliendo però cómpletamente la linea 150 si potranno inserire velocità intermedie come 1.5, 7.3, 4.4 punti al secondo, ecc.

Per utilizzare praticamente il programma del Tasto per telegrafia occorre collegare il circuito di fig. 168 tra l'uscita H dell'Interfaccia e l'ingresso CW del trasmettitore. Come noterete alle linee 230, 250, 280 e 300 è stata usata, invece della POKE 8888 la POKE 9999 che, come già detto all'inizio di questo capitolo, indica la generazione

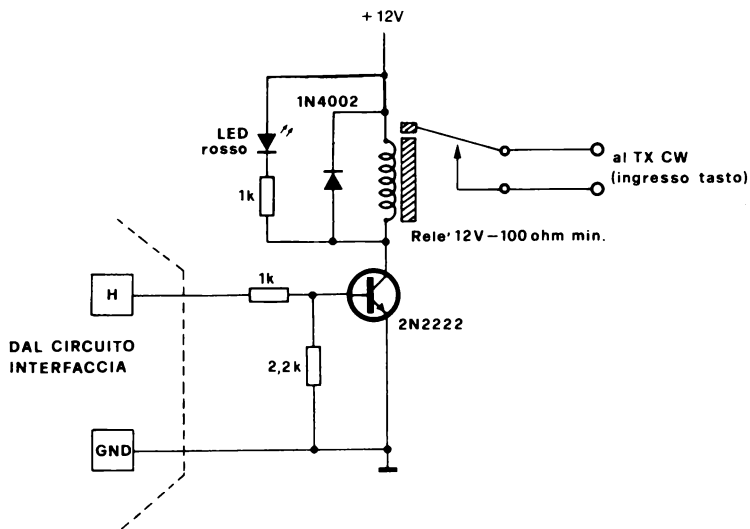



Figura 168

di un suono, questo perchè, collegando all'Interfaccia il circuito musicale descritto nel prossimo capitolo, si può dotare il Tasto per telegrafia di un monitor sonoro senza modificare il programma.

MUSICA

Scheda musicale

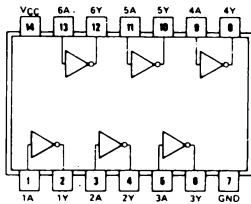
Con una spesa inferiore alle 10000 lire si può collegare ai computers ZX80 e ZX81 (tramite il circuito di Interfaccia già esposto nel capitolo precedente) una scheda musicale capace di produrre 50 note musicali su 4 ottave tramite delle semplici istruzioni in BASIC.
Lo scopo principale è quello di permettere al computer di produrre della musica, però tale scheda si presta a molti altri usi come per esempio sonorizzare dei giochi, produrre dei segnali di allarme, e, comunque, per inserire nei programmi delle note acustiche ovunque può essere utile.
Il circuito (fig. 169) usa tre comuni integrati TTL-LS più un timer 555, un transistor 2N2222 e pochi altri componenti passivi.
Le configurazioni dei quattro integrati e del transistor sono in fig. 170. Il circuito può essere montato su un pezzo di circuito stampato già forato.
È essenziale che le resistenze siano tutte al **5%** (1/4 W), che il condensatore da 4700 pF sia del tipo poliestere al 5 - 10 % massimo che i TTL usati siano della serie **LS** ed infine che i Trimmers siano di buona qualità e possibilmente del tipo incapsulato (come i Trimmer neri e bianchi della PHIER) in modo che la taratura non venga modificata da urti accidentali.
I dieci ingressi (+5V, A.....H, GND) della scheda musicale così realizzata vanno collegati tramite dieci spezzoni di filo di tre colori ai corrispondenti dieci morsetti di uscita del circuito di Interfaccia (fig. 171), fare in particolare attenzione a non invertire tra loro i collegamenti che vanno alle uscite + 5V e GND.
La corrente media che la scheda musicale assorbe dall'alimentatore del computer va dai 30 mA al minimo volume ai 65 mA al massimo volume. Il consumo può essere ridotto e la potenza sonora può essere aumentata togliendo il transistor e l'altoparlante e collegando l'uscita della scheda musicale ad un amplificatore audio più potente ed alimentato esternamente (fig. 172).
La taratura delle 50 frequenze della scheda musicale va effettuata regolando solo i sei trimmers (tm1...tm6) in modo da ottenere determinate frequenze di uscita, il settimo trimmer (tm7) è usato per la regolazione del Volume, vi è anche un interruttore a slitta (SW1) che serve per l'esclusione del suono. Per effettuare la taratura occorre collegare un frequenzimetro digitale all'uscita "DFM" della scheda musicale (vedere fig. 169) e far girare sul computer il seguente programma:

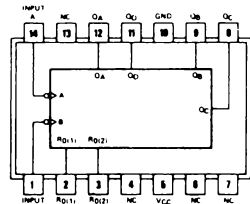
1/4W – 5%
+5V
A
B
C
D
E
F
G
H
GND
DAL CIRCUITO DI INTERFACCIA
1N4148
74LS05
10k
10k
10k
10k
1N4148
trimmers
100k
tm4
tm3
tm2
tm1
trimmer
47k
tm5
15k
tm6
6,8k
22k
trimmer
4,7k
82k
22k
4700p
poliestere
5% !!!
10k
555
10µ
12V
1N4002
SW1
interruttore
esclusione
suono
OFF
ON
8ohm
200
mW
volume
220
tm7
trimmer
con mano
pola
10µ
12V
33
2N2222
1k
2,2k
OUT
D.F.M.
GND
74LS153
+5V
out Y
C0
C1
C2
C3
Aselect
Bselect
Strobe
GND
74LS93
R0
R0
+5V
INA
D
C
B
INB
A
GND

Figura 169

```
5  REM ROUTINE TARATURA
10  FAST
2Ø PRINT "INSERIRE CODICE NOTA"
3Ø INPUT N
4Ø CLS
5Ø POKE 9999,N
6Ø GOTO 2Ø
```

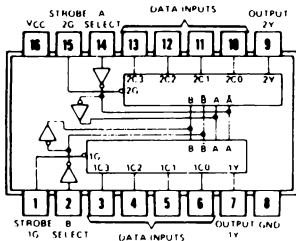
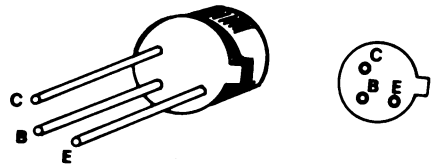
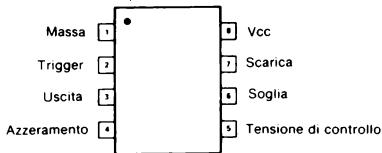
programma necessario per evitare di scrivere ad ogni nota l'istruzione POKE 9999,n.

HEX INVERTERS
WITH OPEN-COLLECTOR OUTPUTS

74LS05

DIVIDE-BY-TWO AND DIVIDE-BY-EIGHT

74LS93

DUAL 4-LINE TO 1-LINE DATA
SELECTORS/MULTIPLEXERS

74LS153

C B E

2N2222

555

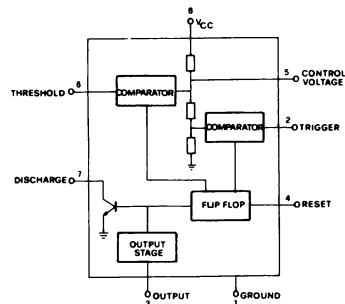



Figura 170

Dopo aver premuto RUN e NEWLINE, le sei tarature vanno effettuate secondo l'ordine indicato nella Tabella 4, introducendo per primo il codice 199 e regolando il trimmer tm1 in modo da avere in uscita una frequenza il più possibile vicina a 523.25 Hz, quindi proseguendo con l'introduzione degli altri cinque codici e con la

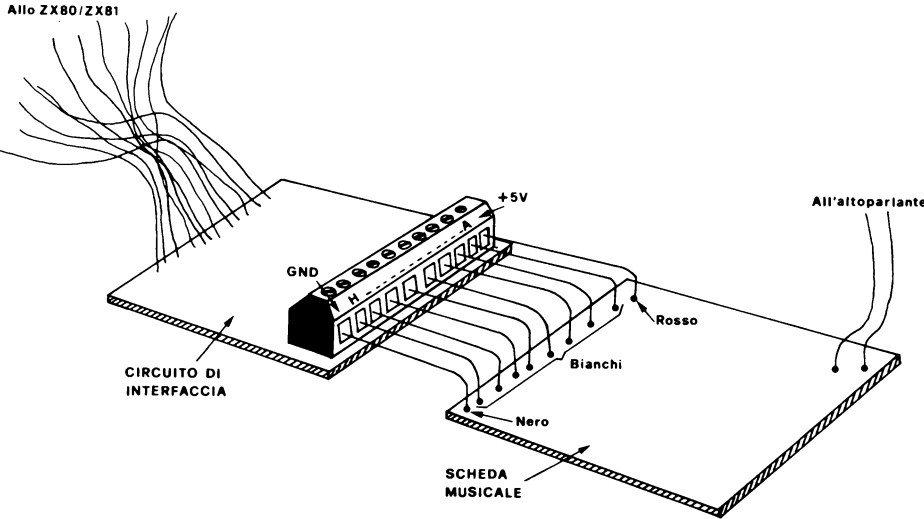



Figura 171

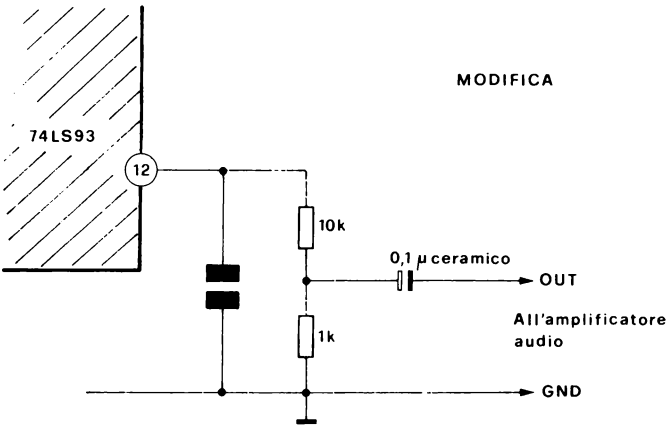



Figura 172

Tabella 4

ORDINE DI TARATURA	POKE 9999, (codice nota)	trimmer da regolare	Frequenza da approssimare (Hz)
1ª	199	tm 1	523,25
2ª	203	tm 2	554,37
3ª	205	tm 3	587,33
4ª	206	tm 4	622,25
5ª	219	tm 5	698,46
6ª	235	tm 6	880,00
tm 7 – Trimmer VOLUME			

regolazione dei corrispondenti trimmers in modo da ottenere le frequenze indicate nella tabella. Nell'effettuare la taratura fare attenzione ad inserire esattamente i sei codici e a non sbagliare nello scegliere ogni volta il trimmer da regolare, un'altra cosa importante è quella di regolare i trimmers con un cacciavite senza urtare gli altri trimmers già tarati e soprattutto senza toccare con le mani i trimmers nè altri

Tabella 5

FREQUENZA DELLA SCHEDA MUSICALE 4 OTTAVE - 50 NOTE				
	NOTA	FREQUENZA STANDARD (Hz)	CODICE NOTA (POKE 9999,)	CODICE ESADECIMALE
1	DO	65,406	7	07
2	DO#	69,296	11	0B
3	RE	73,416	13	0D
4	RE#	77,782	14	0E
	PAUSA	**0**	**15**	**0F**
5	MI	82,407	23	17
6	FA	87,307	27	1B
7	FA#	92,499	29	1D
8	SOL	97,999	30	1E
9	SOL#	103,83	39	27
10	LA	110,00	43	2B
11	LA#	116,54	45	2D
12	SI	123,47	46	2E
13	DO	130,81	71	47
14	DO#	138,59	75	4B

Tabella 5 (Continua)

FREQUENZA DELLA SCHEDA MUSICALE 4 OTTAVE - 50 NOTE				
	NOTA	FREQUENZA STANDARD (Hz)	CODICE NOTA (POKE 9999,)	CODICE ESADECIMALE
15	RE	146,83	77	4D
16	RE#	155,56	78	4E
17	MI	164,81	87	57
18	FA	174,61	91	5B
19	FA#	185,00	93	5D
20	SOL	196,00	94	5E
21	SOL#	207,65	103	67
22	LA	220,00	107	6B
23	LA#	223,08	109	6D
24	SI	246,94	110	6E
25	DO	261,63	135	87
26	DO#	277,18	139	8B
27	RE	293,66	141	8D
28	RE#	311,13	142	8E
29	MI	329,63	151	97
30	FA	349,23	155	9B
31	FA#	369,99	157	9D
32	SOL	392,00	158	9E
33	SOL#	415,30	167	A7
34	LA	440,00	171	AB
35	LA#	466,16	173	AD
36	SI	493,88	174	AE
37	DO	523,25	199	C7
38	DO#	554,37	203	CB
39	RE	587,33	205	CD
40	RE#	622,25	206	CE
41	MI	659,26	215	D7
42	FA	698,46	219	DB
43	FA#	739,99	221	DD
44	SOL	783,99	222	DE
45	SOL#	830,61	231	E7
46	LA	880,00	235	EB
47	LA#	932,33	273	ED
48	SI	987,77	238	EE
49	DO	1046,0	229	E5
50	DO#	1108,7	236	EC

fili o contatti scoperti o piedini degli integrati, questo perchè la resistenza delle dita potrebbe falsare la lettura sul frequenzimetro e di conseguenza la precisione di tutta la taratura. Alla fine della taratura è bene ricontrollare le sei frequenze reinserendo i sei codici della Tabella 4 sempre utilizzando la ROUTINE TARATURA già esposta, routine utile anche successivamente per ascoltare le 50 note che la scheda può produrre.
Le 50 note insieme alle loro frequenze standard e alle corrispondenti POKEs da effettuare per ottenerle sono esposte nella tabella 5, nella stessa Tabella è indicata anche la POKE (9999,15) da effettuare per avere una PAUSA, vale a dire nessuna nota o frequenza emessa.
Se la taratura è stata effettuata con cura la precisione delle 50 note dovrebbe essere migliore dello 0.5 — 1% rispetto ai valori standard indicati nella Tabella 5, il circuito utilizzato per la scheda musicale, oltre alla precisione, ha anche un ottima stabilità sebbene non sia quarzato ma a resistenza e condensatore.
Sempre nella Tabella 5 sono indicati anche i codici esadecimali di tutte le note e della Pausa, codici utili per rendere più compatte all'interno dei programmi le stringhe contenenti lunghe sequenze sonore o musicali.

Programmi

La scheda musicale è senz'altro il circuito più interessante in questo libro per cui ad esso vengono dedicati 4 programmi in questo capitolo sulla musica e ben 14 programmi di effetti sonori nel prossimo capitolo.
Oltre che per la musica e per produrre effetti sonori la scheda musicale può essere usata in molti altri programmi ovunque possa essere necessario o utile aggiungere dei suoni, in questo stesso libro vi sono alcuni esempi, di tali applicazioni nel capitolo prima di questo, dove la scheda è usata come monitor sonoro del Tasto per telegrafia, e nell'ultimo capitolo dove la scheda permette di realizzare dei giochi con anche il sonoro.

Organo a 4 ottave — FAST

Il programma musicale più interessante e certamente quello di fig. 173 che trasforma il computer in un organo elettronico a 4 ottave/49 note da DO 65.4Hz a DO 1046.0 Hz.
L'organo usa la tastiera speciale N. 3 posta alla fine del libro e che va ritagliata e sovrapposta alla tastiera dello ZX. Poichè il computer ha solo 40 tasti, mentre ne occorrerebbero 49, vengono usati solo 25 tasti i quali suonano le 2 ottave più basse premendoli da soli e le 2 ottave più alte premendoli assieme al tasto SHIFT che nella tastiera speciale corrisponde al tasto rosso.
I codici delle POKEs effettuate dal programma, a seconda del tasto premuto e della nota da emettere, sono nella stringa A$ (linea 110). Se la lunghezza (LEN) della

```
  10 REM  ORGANO ELETTRONICO
  20 REM  4 OTTAVE - 26 TASTI
 100 FAST
 110 LET A$="171 199 215 174 015
238 235 015 015 237 222 219 205
229 135 015 015 015 011 014 015
029 039 045 015 015 094 087 078
013 015 093 103 046 109 015 015
110 107 015 015 015 023 075 027
043 091 007 077 030 071 167 157
015 173 231 135 015 151 158 155
139 142 141 203 221 015 206"
 115 PAUSE 30
 120 LET A=CODE INKEY$-12
 130 IF A<1 THEN LET A=5
 140 IF A>51 THEN LET A=A-48
 150 IF A>55 THEN LET A=A-100
 160 LET A=A*4
 170 POKE 9999,VAL A$(A-3 TO A-1
)
 180 GOTO 120
```

Figura 173

stringa A$ è diversa da 271 il programma non funziona, per controllare di avere inserito tutti i codici e gli spazi è sufficiente inserire nel programma la linea 111 (fig. 174), quindi far girare il programma e fermarlo con il tasto BREAK; una volta

```
111 PRINT LEN A$
```

Figura 174

verificato sul video che la lunghezza di A$ è esatta la linea 111 può essere tolta ed il programma utilizzato regolarmente con la tastiera speciale.

Metronomo FAST

Il programma di fig. 175 produce le battute tipiche di un Metronomo il cui numero di battute al minuto (da 60 a 300) può essere predeterminato inserendolo all'inizio (fig. 176).
Il programma può essere fermato usato il tasto BREAK.
La nota emessa dal metronomo può essere cambiata sostituendo il codice 71 della POKE (linea 200) con un altro dei codici della Tabella 5.

```
100 REM  METRONOMO
110 SLOW
120 PRINT "INSERIRE N■ BATTUTE"
130 PRINT "AL MINUTO ( 60--300
)"
140 PRINT "E PREMERE NEWLINE"
150 INPUT B
160 IF B<60 OR B>300 THEN GOTO
150
170 CLS
180 FAST
190 LET T=3000/B-1
200 POKE 9999,71
210 PAUSE 1
220 POKE 9999,15
230 PAUSE T
240 GOTO 200
```

Figura 175

```
INSERIRE N■ BATTUTE
AL MINUTO ( 60--300 )
E PREMERE NEWLINE
```

Figura 176

Scale Musicali FAST

Il programma di fig. 177 esegue alternativamente a salire e a scendere la scala delle 50 note prodotte dalla scheda musicale.

```
1000 REM  SCALE MUSICALI
1010 FAST
1020 LET A$="007 011 013 014 023
 027 029 030 039 043 045 046 071
 075 077 078 087 091 093 094 103
 107 109 110 135 139 141 142 151
 155 157 158 167 171 173 174 199
 203 205 206 215 219 221 222 231
 235 237 238 229 236 "
1030 FOR F=1 TO 200 STEP 4
1040 POKE 9999,VAL A$(F TO F+2)
1050 PAUSE 10
1060 NEXT F
```

Figura 177 (Continua

```
1070 POKE 9999,15
1080 PAUSE 20
1090 FOR F=200 TO 1 STEP -4
1100 POKE 9999,VAL A$(F-3 TO F)
1110 PAUSE 10
1120 NEXT F
1130 POKE 9999,15
1140 PAUSE 20
1150 GOTO 1030
```

Figura 177 (Fine)

La durata di ciascuna nota può essere aumentata o diminuita aumentando o diminuendo le pause alle linee 1Ø5Ø e 111Ø. È anche possibile avere scale solo a salire togliendo le linee da 1Ø9Ø a 114Ø, o scale solo a scendere togliendo le linee da 1Ø3Ø a 1Ø8Ø.
Nella stringa A$ (linea 1Ø2Ø) è importante che dopo l'ultimo codice (236) e prima degli apici vi sia uno spazio.

Generatore di musica casuale - 50 note FAST

L'ultimo programma (fig. 178) di questo capitolo sulla musica produce una sorta di musica casuale formata da combinazioni delle 50 note.

```
1000 REM  GENERATORE DI MUSICA
1005 REM  CASUALE  - 50 NOTE
1010 FAST
1020 LET T=10
1030 LET A$="007 011 013 014 023
 027 029 030 039 043 045 046 071
 075 077 078 087 091 093 094 103
 107 109 110 135 139 141 142 151
 155 157 158 167 171 173 174 199
 203 205 206 215 219 221 222 231
 235 237 238 229 236 "
1040 RAND
1050 FOR N=1 TO 1000
1060 LET F=(1+INT (RND*50))*4
1070 POKE 9999,VAL A$(F-3 TO F)
1080 PAUSE T
1090 IF INKEY$="7" AND T>1 THEN
LET T=T-1
1100 IF INKEY$="6" THEN LET T=T+
1
1110 NEXT N
1120 POKE 9999,15
```

Figura 178

La velocità dell'esecuzione musicale può essere aumentata premendo il tasto "7" e diminuita premendo il tasto "6".
Il programma produce una sequenza di mille note, tale sequenza può essere continua togliendo la linea 1Ø5Ø e sostituendo la linea 111Ø con la linea:
111Ø GOTO 116Ø
Nella stringa A$ (linea 103Ø) è importante che dopo l'ultimo codice (236) e prima degli apici vi sia uno spazio.

EFFETTI SONORI

I programmi di effetti sonori funzionano tutti nel modo FAST e possono essere utilizzati sia da soli che come subroutines di altri programmi. Se avete realizzato e collegato al computer l'Avvisatore acustico per tastiera presentano all'inizio di questo libro, questo va disinserito tramite l'apposito interruttore quando viene fatto girare uno dei programmi di effetti sonori da solo o inserito in altri programmi.

Sirena a 2 toni (fig. 179) **FAST**

```
1000 REM  SIRENA A 2 TONI
1010 FAST
1020 FOR D=1 TO 30
1030 POKE 9999,141
1040 PAUSE 25
1050 POKE 9999,158
1060 PAUSE 25
1070 NEXT D
1080 POKE 9999,15
```

Figura 179

Sirena americana (fig. 180) **FAST**

```
1000 REM  SIRENA AMERICANA
1010 FAST
1020 FOR D=1 TO 90
1030 POKE 9999,199
1035 PAUSE 1
1040 POKE 9999,203
1045 PAUSE 1
1050 POKE 9999,205
1055 PAUSE 1
1060 POKE 9999,206
1065 PAUSE 1
1070 POKE 9999,215
1075 PAUSE 1
1080 POKE 9999,219
```

Figura 180 (Continua)

```
1085 PAUSE 1
1090 POKE 9999,221
1095 PAUSE 1
1100 POKE 9999,222
1105 PAUSE 1
1110 POKE 9999,221
1115 PAUSE 1
1120 POKE 9999,219
1125 PAUSE 1
1130 POKE 9999,215
1135 PAUSE 1
1140 POKE 9999,206
1145 PAUSE 1
1150 POKE 9999,205
1155 PAUSE 1
1160 POKE 9999,203
1165 PAUSE 1
1170 NEXT D
1180 POKE 9999,15
```

Figura 180 (Fine)

Sirena multitono (fig. 181) **FAST**

```
1000 REM  SIRENA MULTITONO
1010 FAST
1020 FOR D=1 TO 30
1030 LET A$="091 103 141 199 "
1040 FOR F=1 TO LEN A$ STEP 4
1050 POKE 9999,VAL A$(F TO F+2)
1060 PAUSE 5
1070 NEXT F
1080 FOR F=LEN A$ TO 1 STEP -4
1090 POKE 9999,VAL A$(F-3 TO F)
1100 PAUSE 5
1110 NEXT F
1120 NEXT D
1130 POKE 9999,15
```

Figura 181

In questa sirena il numero dei toni diversi dipende dal numero di codici contenuti nella stringa A$ (linea 1030). Nell'esempio di fig. 181 vi sono 4 codici di note per cui la sirena è a 4 toni, dopo ogni codice di nota va sempre inserito uno spazio compreso anche l'ultimo codice (nell'esempio, dopo il codice 199).
Altre modifiche possono riguardare le linee 1060 e 1100 che stabiliscono la durata di ogni nota e la linea 1020 che può essere modificata per aumentare o diminuire la durata del suono.

Mitra (fig. 182) FAST

```
1000 REM   MITRA
1010 FAST
1015 RAND
1020 FOR N=1 TO 10
1030 FOR C=1 TO 3+INT (RND*10)
1040 POKE 9999,30
1050 FOR T=1 TO 4
1060 NEXT T
1070 POKE 9999,15
1080 FOR T=1 TO 4
1090 NEXT T
1100 NEXT C
1110 PAUSE 20+INT (RND*30)
1120 NEXT N
```

Figura 182

Campanella arrivo treni in stazione (fig. 183) FAST

```
1000 REM   CAMPANELLA ARRIVO
1005 REM   TRENI IN STAZIONE
1010 FAST
1020 FOR N=1 TO 200
1030 POKE 9999,238
1040 FOR T=1 TO 12
1050 NEXT T
1060 POKE 9999,15
1070 FOR T=1 TO 1
1080 NEXT T
1090 NEXT N
```

Figura 183

Segnale telefonico di linea occupata (fig. 184) FAST

```
1000 REM   SEGNALE TELEFONICO
1005 REM   DI LINEA OCCUPATA
1010 FAST
1020 FOR N=1 TO 30
1030 POKE 9999,215
1040 PAUSE 10
1050 POKE 9999,15
1060 PAUSE 10
1070 NEXT N
```

Figura 184

Segnale telefonico con linea libera (fig. 185) FAST

```
1000 REM  SEGNALE TELEFONICO
1005 REM  DI LINEA LIBERA
1010 FAST
1020 FOR N=1 TO 10
1030 POKE 9999,215
1040 PAUSE 70
1050 POKE 9999,15
1060 PAUSE 70
1070 NEXT N
```

Figura 185

Squillo del telefono (fig. 186) FAST

```
1000 REM  SQUILLO DEL TELEFONO
1010 FAST
1020 FOR N=1 TO 8
1030 FOR F=1 TO 40
1040 POKE 9999,237
1050 FOR T=1 TO 4
1060 NEXT T
1070 POKE 9999,15
1080 NEXT F
1090 PAUSE 80
1100 NEXT N
```

Figura 186

Orologio (fig. 187) FAST

```
1000 REM  OROLOGIO
1010 FAST
1020 FOR N=1 TO 60
1030 POKE 9999,94
1040 FOR T=1 TO 3
1050 NEXT T
1060 POKE 9999,15
1070 PAUSE 25
1080 POKE 9999,71
1090 FOR T=1 TO 3
1100 NEXT T
1110 POKE 9999,15
1120 PAUSE 25
1130 NEXT N
```

Figura 187

Grillo (fig. 188) **FAST**

```
1000 REM  GRILLO
1010 FAST
1020 FOR N=1 TO 50
1030 FOR K=1 TO 5
1040 POKE 9999,240
1050 FOR T=1 TO 2
1060 NEXT T
1070 POKE 9999,15
1080 FOR T=1 TO 2
1090 NEXT T
1100 NEXT K
1110 PAUSE 25
1120 NEXT N
```

Figura 188

Sostituendo nel programma le linee 1030 e 1110 con quelle di fig. 189 si può ottenere un effetto simile a più Grilli.

```
1000 REM  GRILLI

1030 FOR K=1 TO 3

1110 PAUSE 6
```

Figura 189

Il codice 240 usato nella POKE della linea 1040 non è uno dei codici nota della Tabella 5 ma è il codice che fa produrre alla scheda musicale la frequenza più alta (circa 1330 Hz).

Segnale orario (fig. 190) **FAST**

```
1000 REM  SEGNALE ORARIO
1010 FAST
1020 FOR A=1 TO 5
1030 POKE 9999,238
1040 PAUSE 4
1050 POKE 9999,15
1060 PAUSE 46
1070 NEXT A
1080 PAUSE 50
1090 POKE 9999,238
1100 PAUSE 4
1110 POKE 9999,15
```

Figura 190

Sveglia (fig. 191) FAST

```
1000 REM  SVEGLIA
1010 FAST
1020 FOR N=1 TO 200
1030 POKE 9999,222
1040 POKE 9999,231
1050 POKE 9999,235
1060 FOR T=1 TO 4
1070 NEXT T
1080 POKE 9999 15
1090 LET X=1
1100 NEXT N
```

Figura 191

Din - Don (fig. 192) FAST

```
1000 REM  DIN-DON
1010 FAST
1020 FOR N=1 TO 10
1030 POKE 9999,238
1040 PAUSE 2
1050 POKE 9999,15
1060 PAUSE 2
1070 POKE 9999,222
1080 PAUSE 20
1090 POKE 9999,15
1100 PAUSE 50
1110 NEXT N
```

Figura 192

Generatore di suoni casuali (fig. 193) FAST

```
1000 REM  GENERATORE DI SUONI
1005 REM  CASUALI
1010 FAST
1020 PRINT "INSERIRE RITARDO TRA
 SUONO E"
1030 PRINT "SUONO( 1--20 ) E PRE
MERE NEWLINE"
1040 INPUT R
1050 IF R<1 OR R>20 THEN GOTO 10
40
```

Figura 193 (Continua)

```
1060 CLS
1070 RAND
1080 FOR N=1 TO 1000
1090 POKE 9999,RND*255
1100 FOR T=1 TO R
1110 NEXT T
1120 NEXT N
```

Fig. 193 (Fine)

GIOCHI

Prima di far girare il programma giochi che usano la scheda musicale per produrre dei suoni, occorre disinserire l'Avvisatore Acustico per tastiera tramite l'apposito interruttore.

Miniroulette (con sonoro) SLOW

Il primo gioco (fig. 194) è una Roulette con solo 24 posizioni.

```
  10 REM  MINI ROULETTE
 100 SLOW
 105 RAND
 110 PRINT AT 0,19;"MINI ROULETT
E"
 120 FOR A=1 TO 24 STEP 2
 130 PRINT AT 10-10*COS (A/12*PI
),10+10*SIN (A/12*PI);CHR$ (A+27
)
 140 PRINT AT 10-10*COS ((A+1)/1
2*PI),10+10*SIN ((A+1)/12*PI);CH
R$ (A+156)
 150 NEXT A
 160 LET CX=0
 170 LET CY=0
 180 PRINT AT 20,17;"PREMERE NEW
LINE"
 190 PRINT AT 21,17;"PER INIZIAR
E"
 200 INPUT A$
 210 UNPLOT CX,CY
 220 PRINT AT 20,17;"
  "
 230>PRINT AT 21,17;"
 "
 240 LET N=25+INT (RND*25)
 250 FOR H=1 TO N
 260 LET F=H/12*PI
 270 LET CX=21+18*SIN F
```

Figura 194 (Continua)

```
280 LET CY=22.5+17*COS F
290 PLOT CX,CY
300 POKE 9999,199
310 FOR T=1 TO H/3
320 NEXT T
330 POKE 9999,15
340 IF H=N THEN GOTO 180
350 UNPLOT CX,CY
360 NEXT H
```

Figura 194 (Fine)

La Roulette appare sul video come in fig. 195. Premendo NEWLINE la pallina (il quadratino nero all'interno del cerchio) inizia a ruotare in senso orario fermandosi casualmente in una delle 24 posizioni. Durante la rotazione della pallina viene emesso un suono prodotto dalle linee 300 e 330 del programma.
Il programma può essere fermato durante la rotazione della pallina usando il tasto BREAK.
note: La linea 220 contiene tra gli apici 15 spazi, la linea 230 contiene tra gli apici 12 spazi.

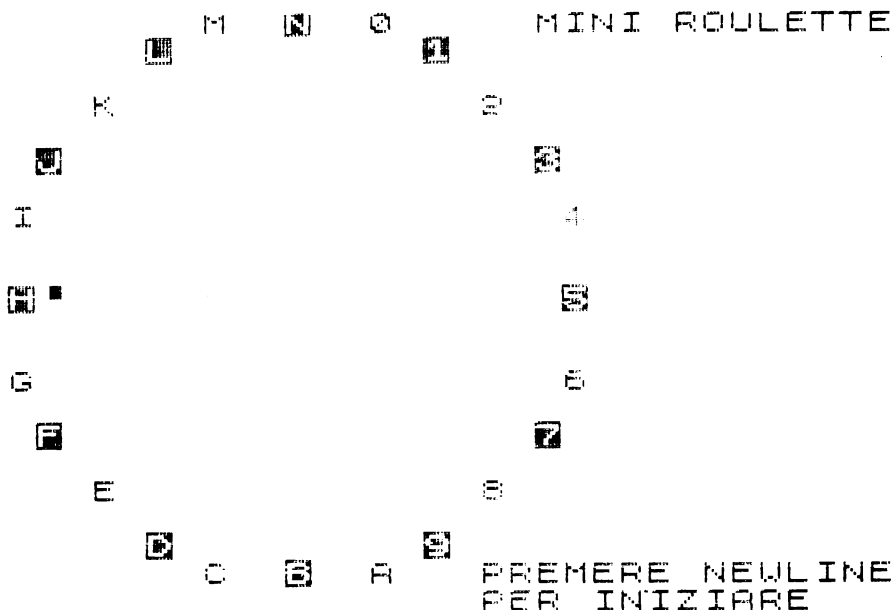



Figura 195

Questo programma (fig. 196) simula un campo in cui vi sono 7 mine nascoste casualmente. Il giocatore deve spostare usando i tasti 5, 6, 7, 8, il punto lampeggiante dalla posizione di partenza nell'angolo in basso a sinistra alla posizione di arrivo nell'angolo in alto a destra senza incappare in una delle mine nascoste. Per attraversare il campo minato senza danni il giocatore può contare sul proprio intuito e sulla propria fortuna, se vi riesce appare sul video la scritta "SALVOOO!!!"

```
  10 REM  CAMPO MINATO
  20 REM ---------------
  99 SLOW
 100 POKE 9999,15
 110 FOR A=1 TO 88
 120 PRINT "▒▒▒▒▒▒▒▒";
 130 NEXT A
 140 PRINT
 150 LET L=21
 160 LET C=0
 170 PRINT AT 0,31;"A"
 180 PRINT AT L,C;"P"
 190 DIM M(7,2)
 200 RAND
 210 FOR A=1 TO 7
 220 LET G=INT (32*RND)
 230 LET F=INT (22*RND)
 240 IF G<12 OR F>14 THEN GOTO 2
20
 250 LET M(A,1)=F
 260 LET M(A,2)=G
 270 NEXT A
 280 PRINT AT L,C;"."
 290 POKE 9999,171
 300 PRINT AT L,C;" "
 310 POKE 9999,15
 320 LET X=L
 330 LET Y=C
 340 IF C>0 AND INKEY$="5" THEN
LET C=C-1
 350 IF L<21 AND INKEY$="6" THEN
 LET L=L+1
 360 IF L>0 AND INKEY$="7" THEN
LET L=L-1
 370 IF C<31 AND INKEY$="8" THEN
 LET C=C+1
 380 IF X<>L OR Y<>C THEN GOTO 4
00
```

Figura 196 (Continua)

```
 390 GOTO 280
 400 FOR A=1 TO 7
 410 IF M(A,1)=L AND M(A,2)=C TH
EN GOTO 450
 420 NEXT A
 430 IF L=0 AND C=31 THEN GOTO 5
30
 440 GOTO 280
 450 FOR E=1 TO 10
 460 PRINT AT L,C;" "
 470 POKE 9999,23*E
 480 PRINT AT L,C;CHR$ (E+139)
 490 POKE 9999,15
 500 NEXT E
 510 PRINT AT L,C;"▚"
 520 STOP
 530 PRINT AT 0,31;"."
 540 PRINT AT 8,0;"
SALVOOO ///
"
```

Figura 196 (Fine)

```
245 PRINT AT F,G;"*"
```

Figura 197

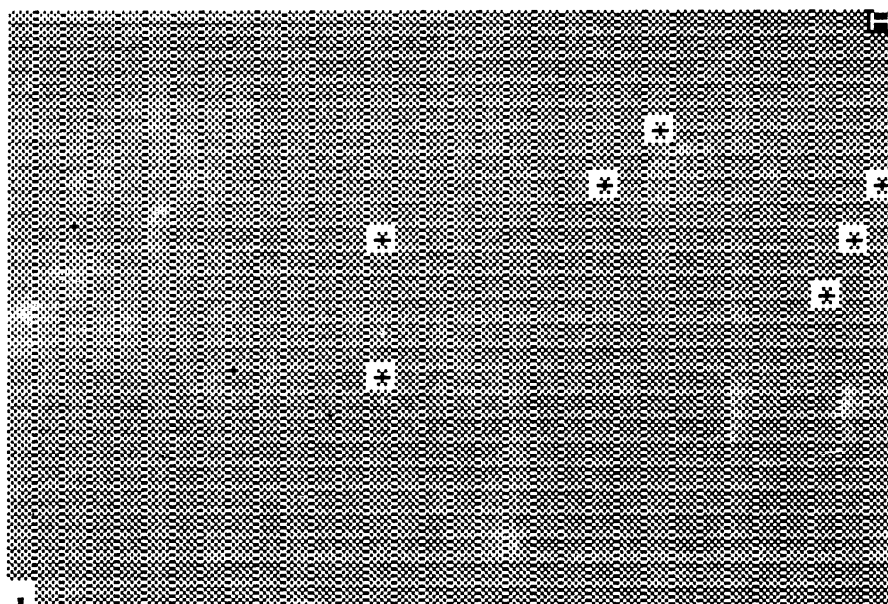



Figura 198

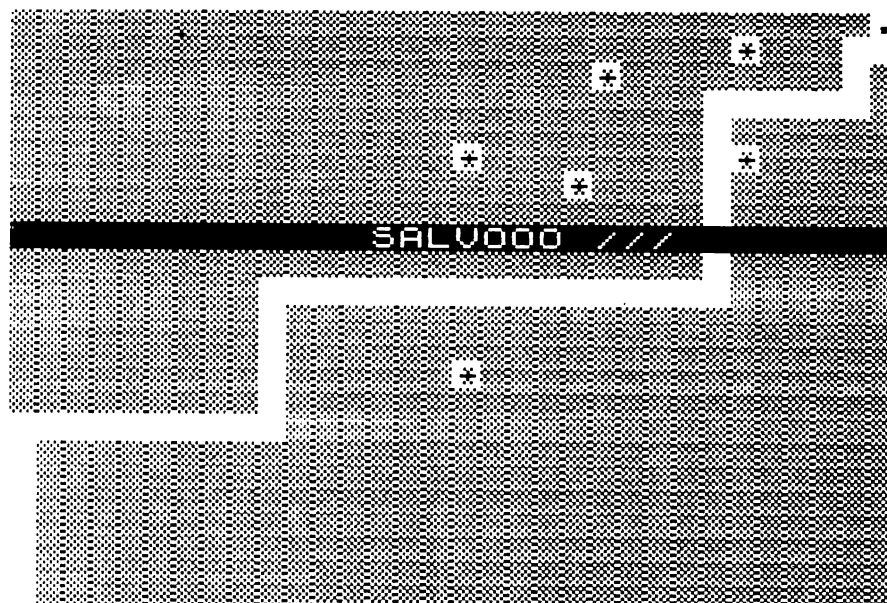



Figura 199

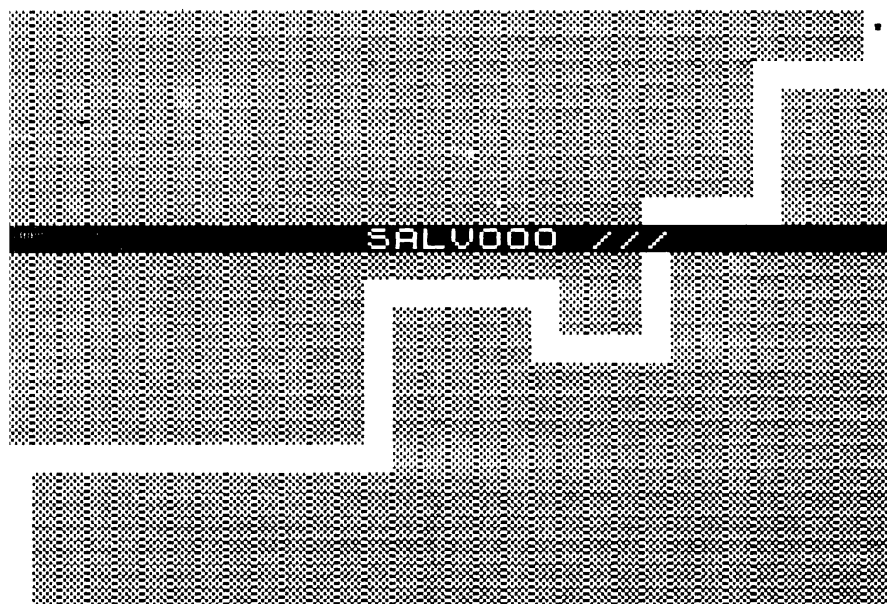



Figura 200

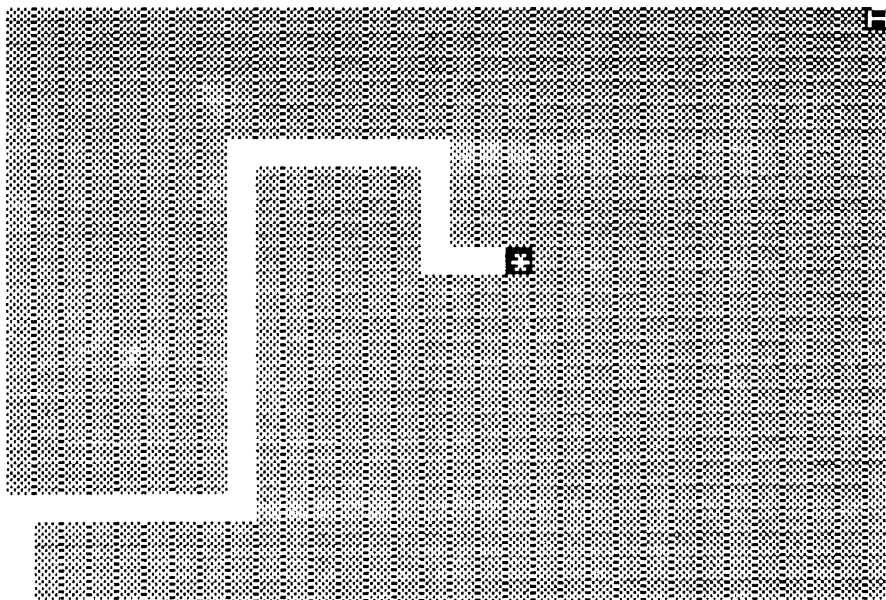



Figura 201

Aggiungendo la linea 245 (fig. 197) e facendo girare il programma si può avere una idea di come vengono disposte le mine sul campo (fig. 198). Se si sposta il punto lampeggiante dalla partenza all'arrivo evitando le mine come già detto appare la scritta "SALVOOO" (fig. 199).

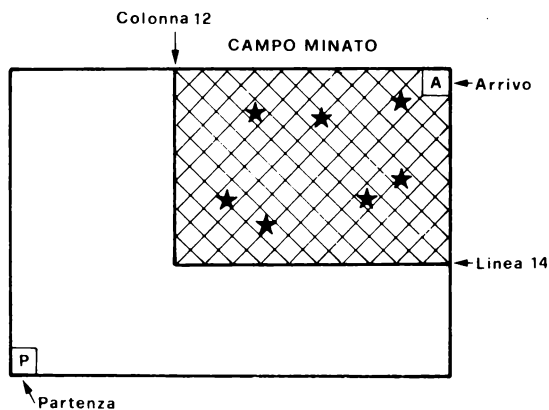



Figura 202

Naturalmente durante il gioco le mine devono essere nascoste per cui si deve prima togliere dal programma la linea 245.
Ora con le mine nascoste può capitare di terminare il gioco senza danno (fig. 200) oppure può succedere prima dell'arrivo di capitare su una delle mine (fig. 201). Se attraversando il campo si finisce su una mina il computer simula con il suono e con l'immagine una sorta di esplosione.
Il numero delle mine può essere aumentato o diminuito per rendere più difficile o più facile ilgioco. Se per esempio si vogliono nascondere 12 mine invece di 7 occorre modificare il contenuto delle linee 190, 210 e 400 sostituendo, all'interno delle linee stesse, tutti i 7 con dei 12.
Così com'è, il programma nasconde le mine in una zona limitata prossima alla posizione di arrivo (fig. 202), se però si modifica il programma in modo da nascondere più mine (per esempio 15-20 mine) si può fare in modo che tali mine vengano sparse per tutto il campo e non in una zona limitata, togliendo dal programma la linea 240.
nota: La linea 120 contiene tra gli apici 8 caratteri grafici ottenuti premendo (con il computer in modo GRAPHIC) insieme al tasto SHIFT i tasti A H A H A- H A H.

1——40 Dadi

FAST O SLOW

Sfruttando al massimo la risoluzione grafica dello ZX81 si possono fare apparire sullo schermo fino a quaranta Dadi, usando il programma di fig. 203. Premendo RUN e NEWLINE il programma chiede innanzitutto quanti dadi devono uscire (fig. 204) quindi li disegna sullo schermo indicando alla fine il numero di Dadi ed il loro punteggio totale (fig. 205).
note: Le linee da 230 a 330 contengono tra gli apici 4 caratteri grafici ciascuna, ottenuti con il computer in modo GRAPHIC e premendo SHIFT. Qui di seguito viene data una lista dei tasti ai quali corrispondono i caratteri grafici da inserire tra gli apici nelle varie linee, in questa lista gli spazi inversi sono indicati con SPACE (da premersi sempre in modo GRAPHIC ma senza SHIFT):

```
23Ø    3, 6, 6, 6
24Ø    8, SPACE, SPACE, SPACE
25Ø    8, Q, SPACE, SPACE
26Ø    8, Q, SPACE, Q
27Ø    8, Q, Q, Q
28Ø    8, SPACE, Q, SPACE
29Ø    8, SPACE, SPACE, SPACE
3ØØ    8, SPACE, SPACE, SPACE
31Ø    8, SPACE, SPACE, Q
32Ø    8, Q, SPACE, Q
33Ø    8, Q, Q, Q,
```

```
  10 REM    1--40 DADI
 100 SLOW
 110 PRINT "INSERIRE N■DADI ( 1-
-40 )"
 120 PRINT "E PREMERE NEWLINE"
 130 INPUT N
 140 IF N<1 OR N>40 THEN GOTO 13
0
 150 CLS
 160 LET K=0
 170 LET Y=0
 180 RAND
 190 FOR L=0 TO 19 STEP 4
 200 FOR C=0 TO 31 STEP 4
 210 LET Y=Y+1
 220 LET X=1+INT (RND*6)
 230 PRINT AT L,C;"████"
 240 IF X=1 THEN PRINT AT L+1,C;
"████"
 250 IF X=2 OR X=3 THEN PRINT AT
 L+1,C;"████"
 260 IF X=4 OR X=5 THEN PRINT AT
 L+1,C;"████"
 270 IF X=6 THEN PRINT AT L+1,C;
"████"
 280 IF X=1 OR X=3 OR X=5 THEN P
RINT AT L+2,C;"████"
 290 IF X=2 OR X=4 OR X=6 THEN P
RINT AT L+2,C;"████"
 300 IF X=1 THEN PRINT AT L+3,C;
"████"
 310 IF X=2 OR X=3 THEN PRINT AT
 L+3,C;"████"
 320 IF X=4 OR X=5 THEN PRINT AT
 L+3,C;"████"
 330 IF X=6 THEN PRINT AT L+3,C;
"████"
 340 LET K=K+X
 350 IF Y=N THEN GOTO 380
 360 NEXT C
 370 NEXT L
 380 PRINT
 390 PRINT "   ";N;" DADI";"    P
UNTEGGIO = ";K
```

Figura 203

```
INSERIRE N■DADI ( 1--40 )
E PREMERE NEWLINE
```

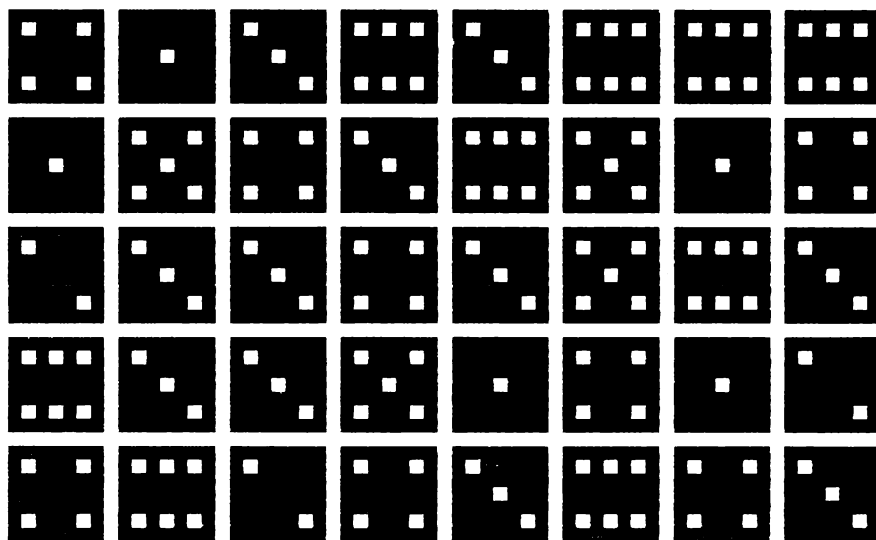



Figura 204

Figura 205

Tavola di numeri e lettere SLOW

La prima parte del programma di fig. 206 disegna sul video una tavola di 16 caselle di cui 15 occupate dai numeri da 0 a 9 più le lettere da A ad E. Subito dopo, con il

```
 10 REM  TAVOLA DI NUMERI
 20 REM  E LETTERE
100 SLOW
110 DIM A$(4,4)
120 LET A$(1)="1234"
130 LET A$(2)="5678"
140 LET A$(3)="90AB"
150 LET A$(4)="CDE "
160 PRINT AT 3,12;"      "
170 PRINT AT 4,12;" 1234 "
180 PRINT AT 5,12;" 5678 "
190 PRINT AT 6,12;" 90AB "
200 PRINT AT 7,12;" CDE  "
210 PRINT AT 8,12;"      "
```

Figura 206 (Continua)

```
 220 FAST
 230 LET L=4
 240 LET C=4
 250 FOR M=1 TO 500
 260 LET X=1+INT (4*RND)
 270 IF C<4 AND X=1 THEN GOSUB 3
80
 280 IF L>1 AND X=2 THEN GOSUB 4
40
 290 IF L<4 AND X=3 THEN GOSUB 5
00
 300 IF C>1 AND X=4 THEN GOSUB 5
60
 310 NEXT M
 320 SLOW
 330 IF C<4 AND INKEY$="5" THEN
GOSUB 380
 340 IF L>1 AND INKEY$="6" THEN
GOSUB 440
 350 IF L<4 AND INKEY$="7" THEN
GOSUB 500
 360 IF C>1 AND INKEY$="8" THEN
GOSUB 560
 370 GOTO 330
 380 PRINT AT L+3,C+13;" "
 390 LET A$(L,C)=A$(L,C+1)
 400 LET A$(L,C+1)=" "
 410 PRINT AT L+3,C+12;A$(L,C)
 420 LET C=C+1
 430 RETURN
 440 PRINT AT L+2,C+12;" "
 450 LET A$(L,C)=A$(L-1,C)
 460 LET A$(L-1,C)=" "
 470 PRINT AT L+3,C+12;A$(L,C)
 480 LET L=L-1
 490 RETURN
 500 PRINT AT L+4,C+12;" "
 510 LET A$(L,C)=A$(L+1,C)
 520 LET A$(L+1,C)=" "
 530 PRINT AT L+3,C+12;A$(L,C)
 540 LET L=L+1
 550 RETURN
 560 PRINT AT L+3,C+11;" "
 570 LET A$(L,C)=A$(L,C-1)
 580 LET A$(L,C-1)=" "
 590 PRINT AT L+3,C+12;A$(L,C)
 600 LET C=C-1
 610 RETURN
```

Figura 206 (Continua)

computer in FAST, le 15 caselle con le lettere e i numeri vengono mescolate per circa 45 secondi per poi apparire sullo schermo come in fig. 207.
A questo punto, usando i tasti 5, 6, 7, 8 si possono spostare le varie caselle con i numeri e le lettere per cercare di ritornare alla disposizione iniziale (fig. 208).

Figura 207

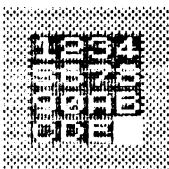



Figura 208

Se all'inizio del programma si vuole vedere in che modo il computer mescola le varie caselle si può sostituire la linea 22Ø FAST con la linea 22Ø SLOW, in questo modo però tale fase di mescolamento durerà circa tre minuti.

Tombola SLOW

Il programma della tombola (fig. 209) per prima cosa disegna in nero sul video le 6 cartelle del tabellone e la scritta "Tombola".

```
  10 REM   TOMBOLA
 100 SLOW
 110 PRINT AT 0,0;""
 120 PRINT AT 1,0;""
 130 FOR L=3 TO 21
 140 IF NOT L=8 AND NOT L=9 AND
NOT L=15 AND NOT L=16 THEN PRINT
 AT L,0;""
 150 NEXT L
```

Figura 209 (Continua)

```
 160 PRINT AT 2,1;"PREMERE ""N""
 PER AVERE UN NUMERO"
 170 PRINT AT 0,16;"▒L▀ULTIMO NU
MERO"
 180 PRINT AT 1,16,"▒USCITO E▀"
 190 DIM A(90)
 200 LET N=0
 210 LET M=0
 220 IF N<0 THEN PRINT AT 1,27;"
>█";CHR$ (N+156);"<"
 230 IF N>9 THEN PRINT AT 1,27;"
>";CHR$ (INT (N/10+156));CHR$ (N
-INT (N/10)*10+156);"<"
 240 LET T=222**2
 250 PRINT AT 1,27;"  ███  "
 260 IF M=90 THEN GOTO 1000
 270 IF INKEY$="N" THEN GOSUB 29
0
 280 GOTO 220
 290 LET M=M+1
 300 RAND
 310 LET X=1+INT (RND*90)
 320 IF A(X)=0 THEN GOTO 360
 330 IF A(X)>0 AND X<90 THEN LET
 X=X+1
 340 IF A(X)>0 AND X=90 THEN LET
 X=1
 350 GOTO 320
 360 LET N=X
 370 LET A(X)=X
 380 IF N<10 THEN LET N$="█"+CHR
$ (N+156)
 390 IF N>9 THEN LET N$=CHR$ INT
 (N/10+156)+CHR$ (N-(INT (N/10)*
10)+156)
 400 IF N<31 THEN LET V=INT ((N-
1)/10+4)
 410 IF N>30 AND N<61 THEN LET V
=INT ((N-1)/10+8)
 420 IF N>60 THEN LET V=INT ((N-
1)/10+12)
 430 LET Y=CODE N$(2 TO )-157
 440 IF Y<5 THEN LET K=Y*3+1
 450 IF Y=-1 THEN LET K=30
 460 IF Y>4 THEN LET K=Y*3+2
 470 PRINT AT V,K;N$
 480 RETURN
 999 GOTO 1000
1000 GOTO 999
```

Figura 209 (Fine)

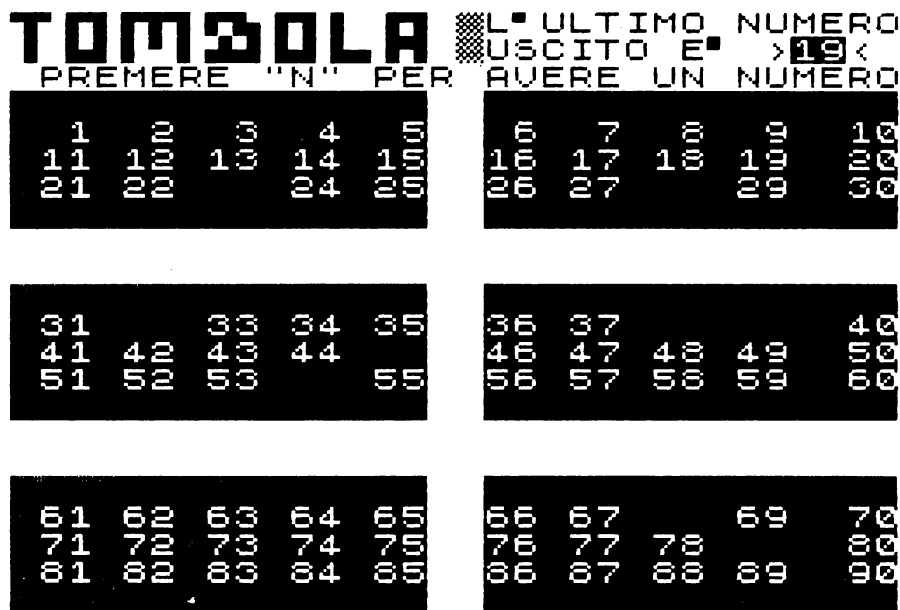



Figura 210

Ora ogni volta che si preme il tasto "N" il programma estrae uno dei numeri da 1 a 90 che non sia già uscito e lo sistema nella sua cartella, inoltre il numero appena uscito lampeggia nell'apposito riquadro nero posto in alto a destra.
È molto importante premere ogni volta il tasto "N" giusto il tempo necessario perchè nel piccolo riquadro lampeggiante appaia un nuovo numero.
L'aspetto del tabellone dopo l'uscita di molti numeri può essere come in fig. 210.
Se si è in molti a giocare si possono collegare più televisori all'uscita del computer e sistemarli in vari punti del tavolo.
Il programma può essere fermato usando il tasto BREAK.

Metal Detector (con sonoro) SLOW

Il programma (fig. 211) disegna un campo e vi nasconde il tesoro in un punto scelto casualmente dal computer.

```
  10 REM   METAL DETECTOR
  20 REM ------------------
 100 SLOW
 110 LET A$="007 011 013 014 023
027 029 030 039 043 045 046 071
075 077 078 087 091 093 094 103
```

Figura 211 (Continua)

```
 107  109  110  135  139  141  142  151
 155  157  158  167  171  173  174  199
 203  222"
 120 LET X=INT (15*RND)+INT (18*
RND)
 130 LET Y=INT (12*RND)+INT (11*
RND)
 140 IF X<8 AND Y>15 THEN GOTO 1
20
 150 LET L=21
 160 LET C=0
 170 FOR S=1 TO 88
 180 PRINT "▒▒▒▒▒▒▒▒";
 190 NEXT S
 200 PRINT
 210 LET M=0
 220 LET F=15
 230 PRINT AT L,C;"."
 240 POKE 9999,15
 250 PRINT AT L,C;"▒"
 260 IF C>0 AND INKEY$="5" THEN
LET C=C-1
 270 IF L<21 AND INKEY$="6" THEN
 LET L=L+1
 280 IF L>0 AND INKEY$="7" THEN
LET L=L-1
 290 IF C<31 AND INKEY$="8" THEN
 LET C=C+1
 300 IF CODE INKEY$>32 AND CODE
INKEY$<37 THEN GOTO 370
 310 IF L=Y AND C=X THEN GOTO 34
0
 320 POKE 9999,F
 330 GOTO 230
 340 PRINT AT Y,X;"■"
 345 POKE 9999,F
 350 PRINT AT 0,0;"TESORO TROVAT
O IN ";M;" MOSSE"
 355 PAUSE 50
 360 POKE 9999,15
 365 STOP
 370 LET M=M+1
 380 LET J=53-ABS (Y-L)-ABS (C-X
)
 385 IF J>13 THEN LET J=J-14
 390 LET F=VAL A$(J*4-3 TO J*4-1
)
 400 GOTO 310
```

Figura 211 (Fine)

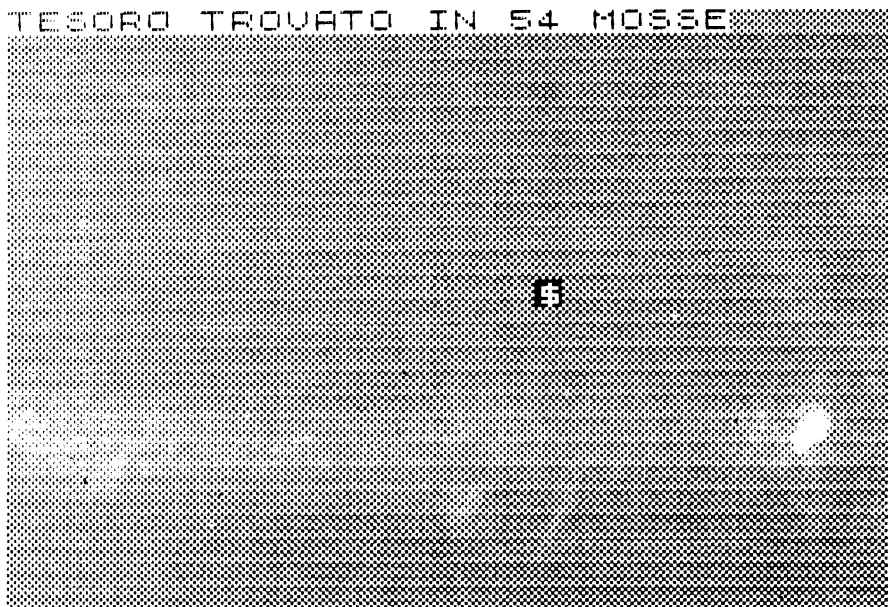



Figura 212

Il giocatore dovrà spostare, tramite i tasti 5, 6, 7, 8, il punto lampeggiante in basso a sinistra fino al punto dove è nascosto il tesoro.
Come in un vero Rivelatore di Metalli la frequenza del suono emessa dal computer aumenterà diventando sempre più acuta quando il punto lampeggiante si avvicina al tesoro nascosto mentre diminuirà se il punto lampeggiante viene spostato nella direzione sbagliata.
Appena il punto lampeggiante passerà sul tesoro apparirà il simbolo del dollaro ed il computer indicherà con una scritta in alto sul video in quanti spostamenti il giocatore è riuscito a trovarlo (fig. 212).

Segnatempo (con sonoro) FAST

Questo programma (fig. 213) non è propriamente un gioco ma può essere utilizzato in molti giochi con limite di tempo per scandire il passare dei secondi.

```
 100 REM  SEGNATEMPO
 110 FAST
 120 PRINT "INSERIRE INTERVALLO
TRA"
 130 PRINT "I SEGNALI  ( 1--30 S
EC. )"
```

Figura 213 (Continua)

```
140 INPUT S
150 IF S<1 OR S>30 THEN GOTO 14
0
160 LET T=S*50-2
170 POKE 9999,7
180 PAUSE 50
190 POKE 9999,15
200 PAUSE T
210 POKE 9999,222
220 PAUSE 2
230 POKE 9999,15
240 GOTO 190
```

Figura 213 (Fine)

Il programma emette un suono basso per un secondo, appena tale suono sparisce il computer comincia ad alternare delle pause e dei brevi BIP sonori.
Il tempo che intercorre tra un BIP e l'altro può essere scelto tra 1 é 30 secondi e va introdotto all'inizio quando il programma lo richiede.

Lotto SLOW

Il programma di fig. 214 estrae dei numeri casuali da giocare al Lotto.
Per prima cosa il programma chiede su quali Ruote si vuole effettuare la Giocata

```
10 REM [illegible]
20 SLOW
100 DIM A$(10,8)
110 LET A$(1)="BARI    "
120 LET A$(2)="CAGLIARI"
130 LET A$(3)="FIRENZE "
140 LET A$(4)="GENOVA  "
150 LET A$(5)="MILANO  "
160 LET A$(6)="NAPOLI  "
170 LET A$(7)="PALERMO "
180 LET A$(8)="ROMA    "
190 LET A$(9)="TORINO  "
200 LET A$(10)="VENEZIA "
210 PRINT "SCEGLIERE LE RUOTE P
REMENDO"
220 PRINT "I TASTI RELATIVI ALL
E CITTA■ "
230 FOR L=5 TO 14
240 PRINT AT L,L;CHR$ (L+151);"
 ";A$(L-4)
```

Figura 214 (Continua)

```
 250 NEXT L
 260 PRINT AT 18,3;"PREMERE <C>
PER CONTINUARE"
 270 DIM H$(1,10)
 280 LET N=CODE INKEY$-28
 290 IF N>=0 AND N<10 THEN LET H
$(1,N+1)="*"
 300 IF N>=0 AND N<10 THEN PRINT
 AT N+5,N+18;"<"
 310 IF N=12 THEN GOTO 330
 320 GOTO 280
 330 CLS
 340 PRINT "SCEGLIERE LA GIOCATA
"
 350 PRINT
 360 PRINT
 370 PRINT "   INSERIRE         PE
R AVERE"
 380 PRINT "----------------------
-------------"
 390 PRINT
 400 PRINT TAB 7;"1";TAB 18;"NUM
ERI SINGOLI"
 410 PRINT
 420 PRINT TAB 7;"2";TAB 18;"AMB
I"
 430 PRINT
 440 PRINT TAB 7;"3";TAB 18;"TER
NI"
 450 PRINT
 460 PRINT TAB 7;"4";TAB 18;"QUA
TERNE"
 470 PRINT
 480 PRINT TAB 7;"5";TAB 18;"CIN
QUINE"
 490 PRINT AT 19,5;"POI PREMERE
NEWLINE"
 500 INPUT G
 510 IF G<1 OR G>5 THEN GOTO 500
 520 CLS
 530 DIM X(5)
 540 LET P=0
 550 FOR T=1 TO 10
 560 IF H$(1,T)="*" THEN GOSUB 6
00
 570 NEXT T
 580 STOP
 600 FOR K=1 TO 5
```

Figura 214 (Continua)

```
610 LET X(K)=1+INT (90*RND)
620 NEXT K
630 IF X(1)=X(2) OR X(1)=X(3) OR X(1)=X(4) OR X(1)=X(5) OR X(2)=X(3) OR X(2)=X(4) OR X(2)=X(5) OR X(3)=X(4) OR X(3)=X(5) OR X(4)=X(5) THEN GOTO 600
640 PRINT AT P,4;A$(T)
650 FOR D=1 TO G
660 PRINT AT P,14+D*3;X(D)
670 NEXT D
680 LET P=P+2
690 RETURN
```

Figura 214 (Fine)

SCEGLIERE LE RUOTE PREMENDO
I TASTI RELATIVI ALLE CITTA

PREMERE <C> PER CONTINUARE

Figura 215

(fig. 215), la scelta delle città viene effettuata premendo i corrispondenti numeri da 1 a Ø (la prima fila di tasti del computer), quindi si preme il tasto "C" per passare alla scelta della Giocata (fig. 216). Tale scelta viene effettuata inserendo un numero da 1 a 5 (per avere un numero singolo, un ambo, ecc.) e premendo poi NEWLINE. A questo punto appaiono sullo schermo le città con a fianco i numeri scelti casualmente dal computer.
La fig. 217 mostra un esempio di risultato ottenuto selezionando tutte le Ruote e scegliendo la cinquina come Giocata.

```
SCEGLIERE LA GIOCATA

   INSERIRE           PER AVERE
--------------------------------
      1           NUMERI SINGOLI
      2           AMBI
      3           TERNI
      4           QUATERNE
      5           CINQUINE


     POI PREMERE NEWLINE
```

Figura 216

BARI	1	8	40	72	25
CAGLIARI	18	19	13	63	7
FIRENZE	60	85	54	51	70
GENOVA	47	56	88	62	3
MILANO	71	14	32	16	68
NAPOLI	88	52	79	23	7
PALERMO	70	17	51	38	55
ROMA	63	90	60	31	25
TORINO	6[illegible]	47	35	57	33
VENEZIA	36	38	82	51	33

Figura 217

Totocalcio

FAST O SLOW

L'ultimo programma di questo libro (fig. 218) visualizza sullo schermo e stampa alla fine sulla stampante delle colonne di risultati da giocare al Totocalcio. Il

```
  10 REM  TOTOCALCIO
  20 REM  GENERAZIONE DI
  30 REM  COLONNE CASUALI
 100 SLOW
 110 PRINT "QUANTE COLONNE ?  (
1--16 )"
 120 INPUT N
 130 IF N<1 OR N>16 THEN GOTO 12
0
 140 CLS
 150 PRINT "INSERIRE PERCENTUALE
 DI USCITA"
 160 PRINT "DEGLI <1> IN %   (
0--100 )"
 170 INPUT P1
 180 IF P1<0 OR P1>100 THEN GOTO
 170
 190 CLS
 200 PRINT "INSERIRE PERCENTUALE
 DI USCITA"
 210 PRINT "DEGLI <X> IN %   (
0--";100-P1;" )"
 220 INPUT PX
 230 IF PX<0 OR PX>(100-P1) THEN
 GOTO 220
 240 CLS
 250 PRINT " 1 -";P1;"%   X -"
;PX;"%   2 -";100-P1-PX;"% "
 260 PRINT
 270 PRINT
 280 RAND
 300 FOR L=1 TO 13
 310 FOR C=1 TO N
 320 LET X=1+INT (100*RND)
 330 IF X<P1+1 THEN PRINT "1  ";
 340 IF X>P1 AND X<(P1+PX+1) THE
N PRINT "X  ";
 350 IF X>(P1+PX) THEN PRINT "2
";
 360 NEXT C
 370 PRINT
 380 IF L=3 OR L=6 OR L=9 THEN P
RINT
 390 NEXT L
 400 COPY
```

Figura 218

numero di colonne desiderato (che può essere da 1 a 16) va introdotto all'inizio del programma (fig. 219).

La scelta dei risultati tra 1, X e 2 è parzialmente casuale poiche il programma prevede all'inizio l'introduzione della percentuale di uscita degli "1" (fig. 220) e degli "X" (fig. 221) che si ritiene più probabile, se la percentuale totale degli "1" e

QUANTE COLONNE ? (1--1[illegible])

Figura 219

INSERIRE PERCENTUALE DI USCITA
DEGLI <1> IN % (0--100)

Figura 220

INSERIRE PERCENTUALE DI USCITA
DEGLI <X> IN % (0--50)

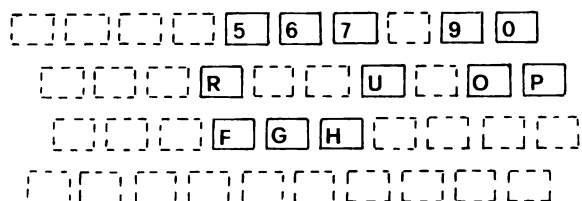
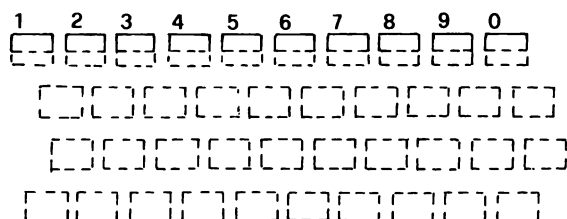
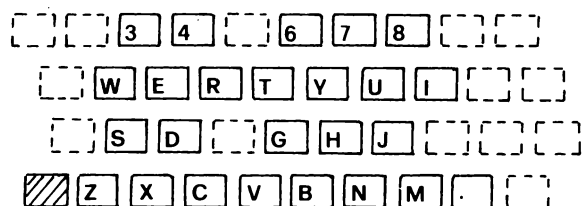
Figura 221

1 ;50 %   X ;30 %   2 ;20 %

```
1 2 2 1 X X 1 1 1 1 1 2 X 1 1 X
X X 2 1 1 X 2 1 1 1 2 1 2 1 1 X
2 2 1 X 1 1 X 2 1 X 1 1 2 1 2 1

X X 1 1 1 X 1 X X X 1 X 2 1 1 X
2 2 1 1 1 X 1 1 1 2 X 2 1 X 1 2
1 X 1 2 2 X 2 1 X X X X 1 X 1 1

2 1 1 X 1 X X X 1 X X 1 X [illegible] X X
1 1 1 X 1 X 1 X 1 1 1 1 1 X X 1
2 X X 2 1 X 1 X X 2 X 1 1 2 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 X X X
1 1 1 1 X 1 X 1 X X X 1 1 1 1 1
2 1 X X 1 X X 1 X 2 X 1 X X X 2
X 1 2 2 X X X X 2 X 1 X X 1 1 1
```

Figura 222

degli "X" e inferiore al 100% la parte rimanente rappresenta la percentuale di uscita del "2".
La fig. 222 mostra un esempio di risultato con 16 colonne e con in alto le percentuali di 1, X, 2, introdotte.

Dove collocare le tastiere allegate

Corrispondenza tra la Tastiera 1 e i tasti dello ZX81

Corrispondenza tra la Tastiera 2 e i tasti dello ZX81

Corrispondenza tra la Tastiera 3 e i tasti dello ZX81

Chi possiede uno ZX81 o anche uno ZX80 con nuova ROM, troverà senz'altro in questo libro molte cose utili.
Innanzitutto 66 programmi che coprono varie applicazioni e le cui principali caratteristiche sono soprattutto il largo uso delle capacità grafiche e di animazione dello ZX81 e (molto importante) la sicurezza di funzionamento dei programmi non essendo stati essi ricopiati dal video ma stampati direttamente dal computer (subito dopo avere girato con successo) con una speciale stampante ad alta qualità, costruita appositamente per questo libro.
Vi sono anche alcuni circuiti collegabili ai due ZX: un alimentatore tampone, un avvisatore acustico per la tastiera e, soprattutto, un'interfaccia per collegare gli ZX ad infinite apparecchiature esterne, oltre ad una scheda musicale che genera ben 50 note su 4 ottave.
E il meglio, in particolare degli ultimi tre circuiti detti, è che possono essere costruiti con POCHE MIGLIAIA di lire ciascuno!
Non solo ma vengono dati, per l'interfaccia e la scheda musicale, anche numerosi programmi applicativi tra cui, specialmente per quest'ultima, molti effetti sonori ed alcuni giochi che sono, probabilmente, gli unici programmi con sonoro disponibili a tutt'oggi per i computers Sinclair.
Ed ancora, altra novità, troverete alcune tastiere disegnate da sovrapporre alla tastiera del Sinclair in alcune applicazioni speciali.

Dello stesso autore:
GLI AMPLIFICATORI DI NORTON QUADRUPLI LM3900 & LM359 con esperimenti.

L. 12.000 Cod. 520 D ISBN-88-7056-123-2

67

GAETANO MARANO

66 PROGRAMMI PER ZX81 E ZX80 CON NUOVA ROM + HARDWARE

GRUPPO EDITORIALE JACKSON